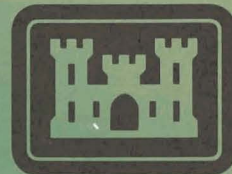


# Environmental & Water Quality Operational Studies



TECHNICAL REPORT E-83-8

## A BOUNDARY-FITTED COORDINATE CODE FOR GENERAL TWO- DIMENSIONAL REGIONS WITH OBSTACLES AND BOUNDARY INTRUSIONS

By Joe F. Thompson

Department of Aerospace Engineering  
Mississippi State University  
Mississippi State, Miss. 39762

March 1983

Final Report

Approved For Public Release; Distribution Unlimited



Prepared for Office, Chief of Engineers, U. S. Army  
Washington, D. C. 20314

Under Contract No. DACW39-78-C-0054  
(EWQOS Task IIIA.4)

Monitored by Hydraulics Laboratory  
U. S. Army Engineer Waterways Experiment Station  
P. O. Box 631, Vicksburg, Miss. 39180



Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
Technical Report E-83-8		
4. TITLE (and Subtitle)		5. TYPE OF REPORT & PERIOD COVERED
A BOUNDARY-FITTED COORDINATE CODE FOR GENERAL TWO-DIMENSIONAL REGIONS WITH OBSTACLES AND BOUNDARY INTRUSIONS		Final report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s)
Joe F. Thompson		Contract No. DACW39-78-C-0054
9. PERFORMING ORGANIZATION NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Mississippi State University Department of Aerospace Engineering Mississippi State, Miss. 39762		EWQOS Task IIIA.4
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE
Office, Chief of Engineers, U. S. Army Washington, D. C. 20314		March 1983
		13. NUMBER OF PAGES
		80
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report)
U. S. Army Engineer Waterways Experiment Station Hydraulics Laboratory P. O. Box 631, Vicksburg, Miss. 39180		Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)		
Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
Available from National Technical Information Service, 5285 Port Royal Road, Springfield, Va. 22151.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
Boundary value problems      Differential equations Computer programs              Finite difference equations Coordinates                      Numerical analysis		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)		
<p>A code (WESCOR) for the generation of boundary-fitted coordinate systems for general two-dimensional regions with boundaries of arbitrary shape and with internal obstacles and boundary intrusions, arbitrary in shape and number, is described and instructions for input and use are given. The coordinate system is generated from the numerical solution of a system of elliptic partial differential equations with provision for controlling the spacing of the coordinate lines in the field. The transformed (computational) region is</p> <p style="text-align: right;">(Continued)</p>		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. ABSTRACT (Continued).

rectangular with the obstacles and intrusions transformed to slits and/or slabs. A small code to distribute points on various fundamental curves with exponential concentration is also described. This front-end code can be used to construct boundary point distributions for input to the coordinate code. A plot code for the coordinate system is also included. The boundary-fitted coordinate systems generated by this code may be used as a basis for the numerical solution of partial differential equations for any physical problem of interest. This procedure will be particularly useful where numerical models are used to analyze flow problems with complex boundary conditions. Typical examples are riverine analysis or the design and evaluation of selective withdrawal outlet works.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

## PREFACE

The developments reported herein were sponsored by the Office, Chief of Engineers, U. S. Army, under the Environmental and Water Quality Operational Studies (EWQOS) Program, through the work unit (CWIS No. 31604) entitled "Techniques to Meet Environmental Quality Objectives for Reservoir Releases." The effort was supported by Task IIIA.4, "Selective Withdrawal Characteristics of Various Outlet Configurations."

This project was conducted during the period June 1978 to May 1982 as a part of a Contract No. DACW39-78-C-0054 with Mississippi State University (MSU) to develop a boundary-fitted coordinate numerical hydrodynamic code for reservoir selective withdrawal. The developments reported herein were conducted by Dr. Joe F. Thompson of the Department of Aerospace Engineering, MSU. This report was written by Dr. Thompson.

Mr. Mark S. Dortch of the Reservoir Water Quality Branch (RWQB), Hydraulic Structures Division, and Dr. Billy H. Johnson of the Mathematical Modeling Group, both within the Hydraulics Laboratory, U. S. Army Engineer Waterways Experiment Station (WES), monitored the contract. Program Manager of EWQOS was Dr. Jerome L. Mahloch, WES Environmental Laboratory. Dr. John Harrison was Chief, EL.

Commanders and Directors of WES during this effort were COL John L. Cannon, CE, COL Nelson P. Conover, CE, and COL Tilford C. Creel, CE. Mr. F. R. Brown was Technical Director.

This report should be cited as follows:

Thompson, J. F. 1983. "A Boundary-Fitted Coordinate Code for General Two-Dimensional Regions with Obstacles and Boundary Intrusions," Technical Report E-83-8, prepared by Mississippi State University, Mississippi State, Miss., for the U. S. Army Engineer Waterways Experiment Station, CE, Vicksburg, Miss.

# Table of Contents

	<u>Page</u>
PREFACE . . . . .	1
INTRODUCTION . . . . .	5
PART A--ELLIPTIC GENERATION SYSTEM . . . . .	8
ELLIPTIC GENERATION SYSTEM . . . . .	8
<u>Basic Ideas</u> . . . . .	8
<u>Mathematical Development</u> . . . . .	16
<u>Types of Boundary-Fitted Coordinate Systems</u> . . . . .	21
<u>Data Required for Generation of Boundary-Fitted</u> <u>Coordinates</u> . . . . .	28
<u>Computer Time Required for Generation of Boundary-</u> <u>Fitted Coordinates</u> . . . . .	31
COORDINATE SYSTEM CONTROL . . . . .	32
<u>Attraction to Other Coordinate Lines and/or Points</u> . . . . .	32
<u>Attraction to Space Curves and/or Points</u> . . . . .	34
<u>Control Functions from Boundary-Point Distributions</u> . . . . .	37
SYSTEM CONFIGURATION . . . . .	38
REFERENCES . . . . .	48
PART B--COORDINATE CODE . . . . .	49
WESCOR (Coordinate System) . . . . .	49
<u>Boundary Configurations</u> . . . . .	50
<u>Control Functions</u> . . . . .	57
<u>Iterative Solution</u> . . . . .	58
<u>Code Operation</u> . . . . .	60

# Table of Contents (continued)

	<u>Page</u>
LINES (Boundary Segments) . . . . .	66
<u>Generation of Curves</u> . . . . .	67
<u>Exponential Concentration of Points</u> . . . . .	70
CSPLØT (Plot) . . . . .	71
WESCØR INPUT INSTRUCTIONS . . . . .	72
LINES INPUT INSTRUCTIONS . . . . .	76
CSPLØT INPUT INSTRUCTIONS . . . . .	77
SAMPLE RUNSTREAMS . . . . .	78

A BOUNDARY-FITTED COORDINATE CODE FOR GENERAL TWO-DIMENSIONAL  
REGIONS WITH OBSTACLES AND BOUNDARY INTRUSIONS

## INTRODUCTION

The use of numerically generated boundary-fitted curvilinear coordinate systems as the basis for numerical solution of partial differential equations on arbitrary regions is now well established. A comprehensive survey of the generation and use of these coordinate systems has recently appeared, Ref. [1], and the proceedings of a recent symposium devoted to this area, Ref. [2], cover the basic techniques involved, as well as applications in many areas.

Such coordinate systems have the property that some coordinate line is coincident with each segment of the boundary in the physical region, so that the complication of boundary shape is effectively removed from the problem. In the past decade the numerical generation of curvilinear coordinate systems has provided the key to the development of finite difference solutions of partial differential equations on regions with arbitrarily shaped boundaries. Although much of the impetus for these developments has come from fluid dynamics, the techniques are equally applicable to heat transfer, electromagnetics, structures, and all other areas involving field solutions.

With coordinate systems generated to maintain coordinate lines (surfaces in 3D) coincident with the boundaries, finite difference codes can be written which are applicable to general configurations without the need of special procedures at the boundaries. Even when the boundaries are in motion, the use of such coordinate systems allows all



computation to be done on a fixed grid with a uniform square mesh in the transformed plane. This greatly simplifies the coding, particularly with regard to boundary conditions, which can now be represented without need of interpolation. It is also possible to distribute the curvilinear coordinate lines in the physical plane with concentration of lines in regions of high gradients while maintaining the square grid in the transformed (computational) plane.

With such systems, the grid points may be thought of as a finite set of observers of the physical solution, stationed so as to be most effective in covering all of the action on the field. The structure of an intersecting net of families of coordinate lines allows the observers to be readily identified in relation to each other. This results in much more simple coding than would the use of a triangular structure or a random distribution of points. The grid generation system provides some influence of each observer on the others so that when one moves to get into a better position, its neighbors will follow in order to maintain smooth coverage of the field. The curvilinear coordinate system thus should cover the field, with coordinate lines (surfaces) coincident with all boundaries. The distribution of lines should be smooth, with concentration in regions of high gradient.

Numerical solutions of partial differential equations are done on the curvilinear coordinate system by first transforming all partial derivatives (or integrals) analytically so that the curvilinear coordinates, rather than the physical coordinates, become the independent variables. Normal and tangential derivatives at boundaries are similarly transformed. (These transformation relations are given in Ref. [3].) The result is a

set of partial differential equations and boundary conditions in which all derivatives (and integrals) are with respect to the curvilinear coordinates. These equations may then be expressed as difference equations on the square grid that is inherent in the transformed plane. There is thus no need for interpolation regardless of the shape of the boundaries or the distribution of the curvilinear coordinate lines in the field.

The present report concerns a code for the generation of boundary-fitted coordinate systems for general 2D regions with boundaries of arbitrary shape and with internal obstacles and boundary intrusions, arbitrary in shape and number. The code, referred to as WESCØR, is described and instructions for input and use are given. Examples of the application of this code are given in Ref. [4]-[6]. The coordinate system is generated from the numerical solution of a system of elliptic partial differential equations with provision for controlling the spacing of the coordinate lines in the field. The transformed (computational) region is rectangular with the obstacles and intrusions transformed to slits and/or slabs. (This type of transformed configuration and its use are discussed in Ref. [3].) A small code to distribute points on various fundamental curves with exponential concentration is also described. This front-end code can be used to construct boundary point distributions for input to the coordinate code. A plot code for the coordinate system is also included. The boundary-fitted coordinate systems generated by this code may be used as a basis for the numerical solution of partial differential equations for any physical problem of interest.

The elliptic generation system is discussed in Part A, and the operation and use of the codes are covered in Part B.

PART A  
ELLIPTIC GENERATION SYSTEM

ELLIPTIC GENERATION SYSTEM

The generation of boundary-fitted coordinates from elliptic systems and the use thereof in the numerical solution of the Navier-Stokes equations is surveyed in Ref. [1]. The foundations of elliptic generation systems are discussed in detail in Ref. [7], and basic configurations of the transformed plane are covered in Ref. [3]. The discussion in this section is an introduction to the subject given by Johnson and Thompson in Ref. [5] and is incorporated here for convenience.

Basic Ideas

Suppose one is interested in solving a differential system involving two concentric circles, such as shown in Fig. 1, where  $r = \text{constant} = \eta_1$  on the inner circle and  $r = \text{constant} = \eta_2$  on the outer circle, and  $\theta$  varies monotonically over the same range over both the inner and outer boundaries, i.e.,  $0^\circ$  to  $360^\circ$ .

A cylindrical coordinate system is the obvious choice since a coordinate line, i.e., a line of constant radius, coincides with each boundary. If one now pulls the interior regions between the two circles

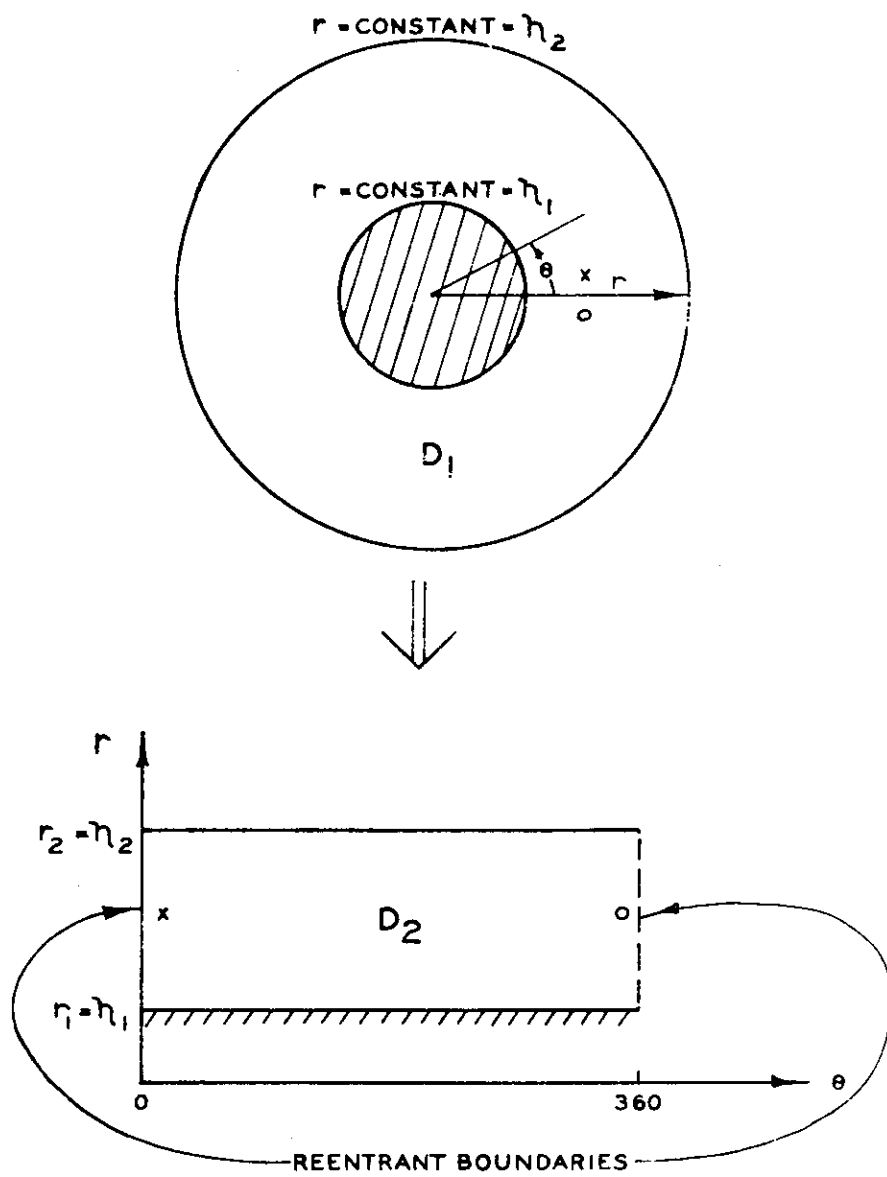
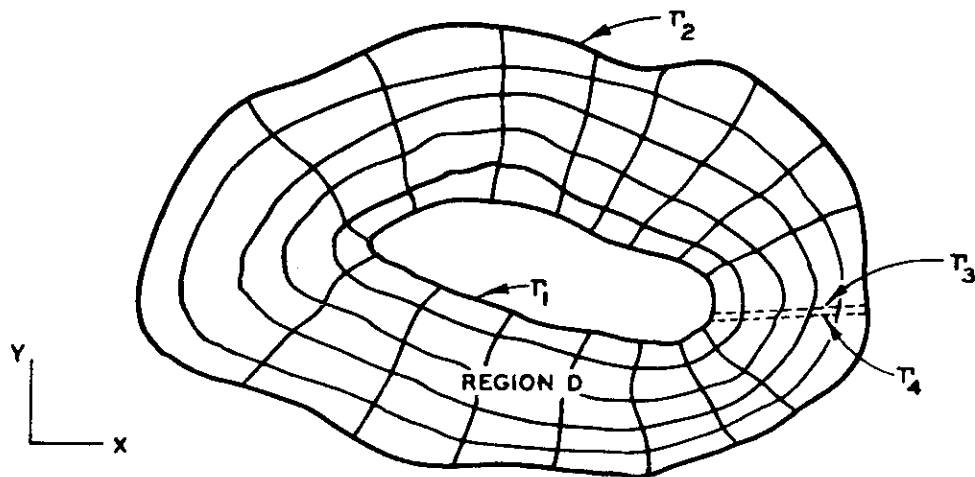


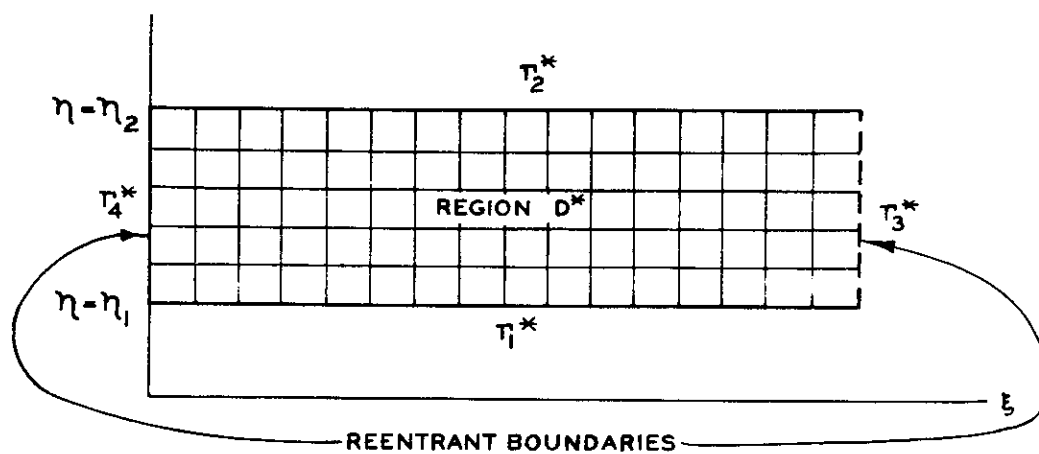
Figure 1. Transformation of domain between concentric cylinders

apart at  $\theta = 0^\circ$  (or  $\theta = 360^\circ$ ) and folds outward, it is easy to visualize the region  $D_1$  becoming the rectangular region  $D_2$ . Likewise, it should be obvious that the right and left sides of the rectangle are reentrant boundaries since  $\theta = 0^\circ$  and  $\theta = 360^\circ$  are coincident in region  $D_1$ . If one computes a derivative in the cylindrical system at  $\theta = 0^\circ$ , values at the points marked x and o on both sides might be used. Thus, these same points, as shown in the rectangular region, would be used for a similar derivative in region  $D_2$ . This is the reason for calling these boundaries reentrant boundaries. As shown, the boundary of the inner circle becomes the bottom of the rectangular region while the boundary of the outer circle becomes the top.

The general boundary-fitted system is completely analogous to the system discussed above. In Fig. 2 the curvilinear coordinate,  $\eta$ , is defined to be constant on the inner boundary in the same way that the curvilinear coordinate,  $r$ , is defined to be constant on the inner circle in the cylindrical coordinate system. Similarly,  $\eta$  is defined to be constant at a different value on the outer boundary. The other curvilinear coordinate,  $\xi$ , is defined to vary monotonically over the same range on both the inner and outer boundaries, as the curvilinear coordinate,  $\theta$ , varies from 0 to  $2\pi$  around both the inner and outer circles in cylindrical coordinates. It would be just as meaningless to have a different range for  $\xi$  on the inner and outer boundaries as it would be to have  $\theta$  increase by something other than  $2\pi$  around one of the circles in cylindrical coordinates. It is this fact that  $\xi$  has the same range on both boundaries that causes the transformed field to be rectangular. Note that the actual values of the coordinates,  $\eta$  and  $\xi$ , are irrelevant,



### PHYSICAL PLANE



### TRANSFORMED PLANE

Figure 2. Transformation of an irregular domain

in the same way that  $r$  and  $\theta$  may be expressed in different units in cylindrical coordinates.

Now that the values of the coordinates,  $\eta$  and  $\xi$ , have been completely specified on all the boundaries of a closed field, it remains to define the values in the interior of the field in terms of these boundary values. Such a task immediately calls to mind elliptic partial differential equations, since the solution of such an equation is completely defined in the interior of a region by its values on the boundary of the region. Thus if the coordinates  $\xi$  and  $\eta$  are taken as the solutions of any two elliptic partial differential equations, say  $L(\xi) = 0$ ,  $D(\eta) = 0$ , where  $L$  and  $D$  represent elliptic operators, then  $\xi$  and  $\eta$  will be determined at each point in the interior of the field by the specified values on the boundary. One condition must be put on the elliptic system chosen, since the same pair of values  $(\xi, \eta)$  must not occur at more than one point in the field or the coordinate system will be ambiguous. This condition can be met by choosing elliptic partial differential equations exhibiting extremum principles that preclude the occurrence of extrema in the interior of the field.

This may be illustrated with resort to the governing equation for a stretched membrane. Consider a membrane attached to a flat plate around a closed circuit of arbitrary shape as shown in Fig. 3. Now let a cylinder of arbitrary flat cross section be pushed up through the plate, stretching the membrane upward. The vertical displacement,  $h$ , of the membrane will be described by Laplace's equation,  $\nabla^2 h = 0$ , with  $h = h_1$  and  $h_2$ , respectively, on the circuits of contact with the plate and cylinder. If equally spaced grid lines encircling the cylinder had been

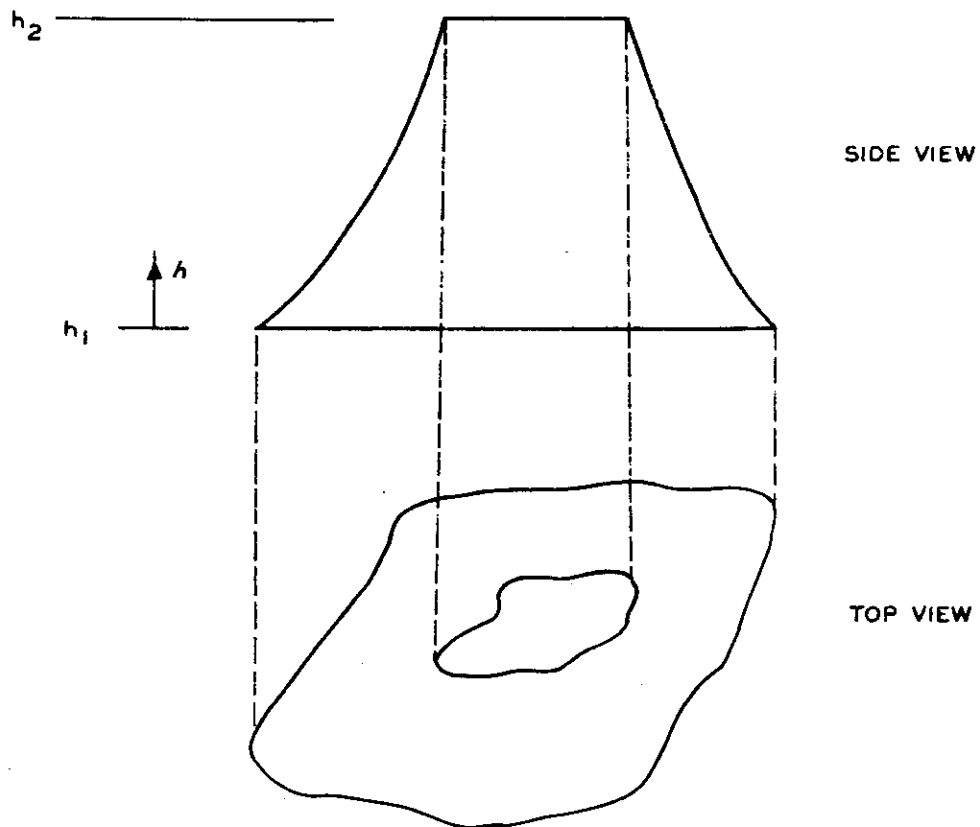


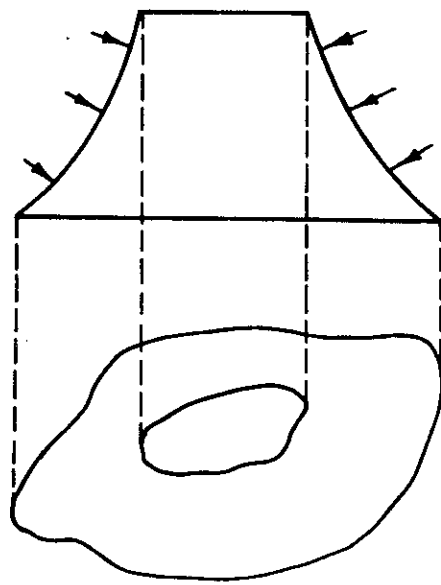
Figure 3. Illustration of extremum principle for Laplace's equation



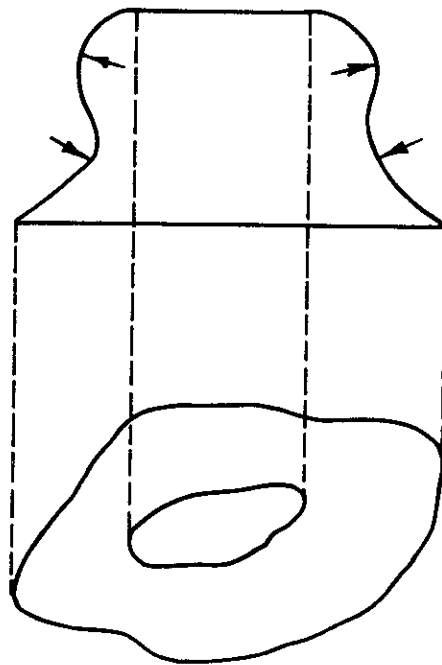
drawn on the membrane before displacement, these lines would appear to move closer to the cylinder when viewed from above after displacement of the membrane. None of these lines would cross, however.

Now let pressure be applied on the upper side of the membrane as diagrammed in Fig. 4a. This will cause the slope at the cylinder to steepen, with the effect that the lines will appear to be drawn even closer to the cylinder but still without crossing. This situation corresponds to the Poisson equation,  $\nabla^2 h = p$ , where  $p$  is the applied pressure. If a variable pressure is applied on both sides of the membrane to a sufficient degree, it is possible to make the membrane assume an S-shape as shown in Fig. 4b. In this case the encircling lines have crossed, and, consequently, a point on the plate can no longer be identified by specifying the encircling line that it lies below (together with a radial ray). This latter case corresponds to a right-hand side of the Poisson equation that is not of one sign over the entire membrane, in which case the extremum principles of Poisson's equation are lost.

Note, however, that if the differential pressure that is applied across the membrane is not too large, the S-shape will not be reached. In this case the lines do not cross, but rather the lines seem to concentrate near a line in the interior of the field. Thus the existence of an extremum principle is a sufficient condition to prevent double-valuedness in the coordinate system but is not a necessary condition. Care must be exercised in its absence, however.



a.



b.

Figure 4. Illustration of extremum principle for Poisson's equation

### Mathematical Development

From the discussion above, a logical choice of the elliptic generating system is Poisson's equation. Thus, based upon Fig. 2, the basic problem is to solve

$$\xi_{xx} + \xi_{yy} = P \quad (1)$$

$$\eta_{xx} + \eta_{yy} = Q$$

with boundary conditions,

$$\begin{aligned} \xi &= \xi_1(x,y) \text{ on } \Gamma_1 \\ \eta &= \text{constant} = \eta_1 \text{ on } \Gamma_1 \end{aligned} \quad (2)$$

$$\xi = \xi_2(x,y) \text{ on } \Gamma_2$$

$$\eta = \text{constant} = \eta_2 \text{ on } \Gamma_2$$

The arbitrary curve joining  $\Gamma_1$  and  $\Gamma_2$  in the physical plane specifies a branch cut for the multiple-valued function,  $\xi(x,y)$ . Thus the values of the coordinate functions  $x(\xi,\eta)$  and  $y(\xi,\eta)$  coincide along  $\Gamma_3$  and  $\Gamma_4$ , and these functions and their derivatives are continuous from  $\Gamma_3$  to  $\Gamma_4$ . Therefore boundary conditions are neither required nor allowed on  $\Gamma_3$  and  $\Gamma_4$ . As previously noted, boundaries with these properties are designated reentrant boundaries.

The functions  $P$  and  $Q$  may be chosen to cause the coordinate lines to concentrate as desired, in analogy with the membrane discussed above.

As discussed in Ref. [7], negative values of  $Q$  result in a superharmonic solution and cause  $\eta$ -lines to move toward the  $\eta$ -line having the lowest value of  $\eta$ , while positive values have the opposite effect. Considering the  $\xi$  solution to be superharmonic results in the interior of the  $\xi =$  constant lines being rotated in a counterclockwise direction in the physical plane; whereas if the  $\xi$ -equation is subharmonic, i.e.,  $P$  is positive, the lines are rotated in the clockwise direction. These effects are discussed in more detail below. It has been found convenient, as discussed in Ref. [7], to redefine the control functions as

$$P = \frac{1}{J^2} (x_\eta^2 + y_\eta^2) P$$

$$Q = \frac{1}{J^2} (x_\xi^2 + y_\xi^2) Q$$

A major purpose of this coordinate system control is to concentrate lines in viscous boundary layers near solid surfaces, and some automated procedures for this purpose have been developed (cf. Ref. [7]). Control is also useful to improve grid spacing and configuration when complicated geometries are involved.

Since all numerical computations are to be performed in the rectangular transformed plane, it is necessary to interchange the dependent and independent variables in Eq. (1). Using the relations given in Ref. [3], Eq. (1) becomes

$$\begin{aligned} \alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} + \alpha P x_\xi + \gamma Q x_\eta &= 0 \\ \alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} + \alpha P y_\xi + \gamma Q y_\eta &= 0 \end{aligned} \tag{3}$$

where

$$\alpha = x_{\eta}^2 + y_{\eta}^2$$

$$\beta = x_{\xi} x_{\eta} + y_{\xi} y_{\eta}$$

$$\gamma = x_{\xi}^2 + y_{\xi}^2$$

$$J = \text{Jacobian of the transformation} = x_{\xi} y_{\eta} - x_{\eta} y_{\xi}$$

with the transformed boundary conditions

$$x = f_1(\xi, \eta_1) \text{ on } \Gamma_1^*$$

$$y = g_1(\xi, \eta_1) \text{ on } \Gamma_1^*$$

$$x = f_2(\xi, \eta_2) \text{ on } \Gamma_2^*$$

$$y = g_2(\xi, \eta_2) \text{ on } \Gamma_2^*$$

Again considering Fig. 2, the boundary functions  $f_1$ ,  $f_2$ ,  $g_1$ , and  $g_2$  are specified by the known shape of the contours  $\Gamma_1$  and  $\Gamma_2$  and the specified distribution of  $\xi$  thereon. Boundary data are neither required nor allowed along the reentrant boundaries  $\Gamma_3$  and  $\Gamma_4$ . Although the new system of equations is more complex than the original system, the boundary conditions are specified on straight boundaries and the coordinate spacing in the transformed plane is uniform. Computationally, these advantages far outweigh any disadvantages resulting from the extra complexity of the equations to be solved.

The boundary-fitted coordinate system so generated has a constant  $\eta$ -line coincident with each boundary in the physical plane. The  $\xi$ -lines may be spaced in any manner desired around the boundaries by specification of  $x,y$  at the equispaced  $\xi$ -points on the  $\Gamma_1^*$  and  $\Gamma_2^*$  lines of the transformed plane. As noted above, the entire side boundaries are reentrant boundaries, and thus neither require nor allow specification of  $x,y$  thereon.

Now the rectangular transformed grid is set up to be the size desired for a particular problem. Since the values of  $\xi$  and  $\eta$  are meaningless in the transformed plane, the  $\eta$ -lines are assumed to run from 1 to the number of  $\eta$ -lines desired in the physical plane. Likewise, the  $\xi$ -lines are numbered 1 to the number specified on the boundaries of the physical plane. The grid spacing in both the  $\xi$  and  $\eta$  directions of the transformed plane is taken as unity. Second-order central difference expressions are used to approximate all derivatives.

Only one of a pair of reentrant boundaries is considered as a computation line since the  $(x,y)$  are equal on both. As an example of how a reentrant boundary is handled, consider the grid in Fig. 5 where "o" indicates a computation point and " $\Delta$ " a boundary point. The derivative of  $x$  with respect to  $\xi$  along  $i = 1$  would be written as

$$\left. \frac{\partial x}{\partial \xi} \right|_{1,j} = (x_{2,j} - x_{\text{IMAX}-1,j})/2 \quad (4)$$

Again, it should be stressed that all computations are performed on the rectangular field with square mesh in the transformed plane. The resulting set of nonlinear difference equations, two for each point, is solved by accelerated Gauss-Seidel (SOR) iteration using overrelaxation.

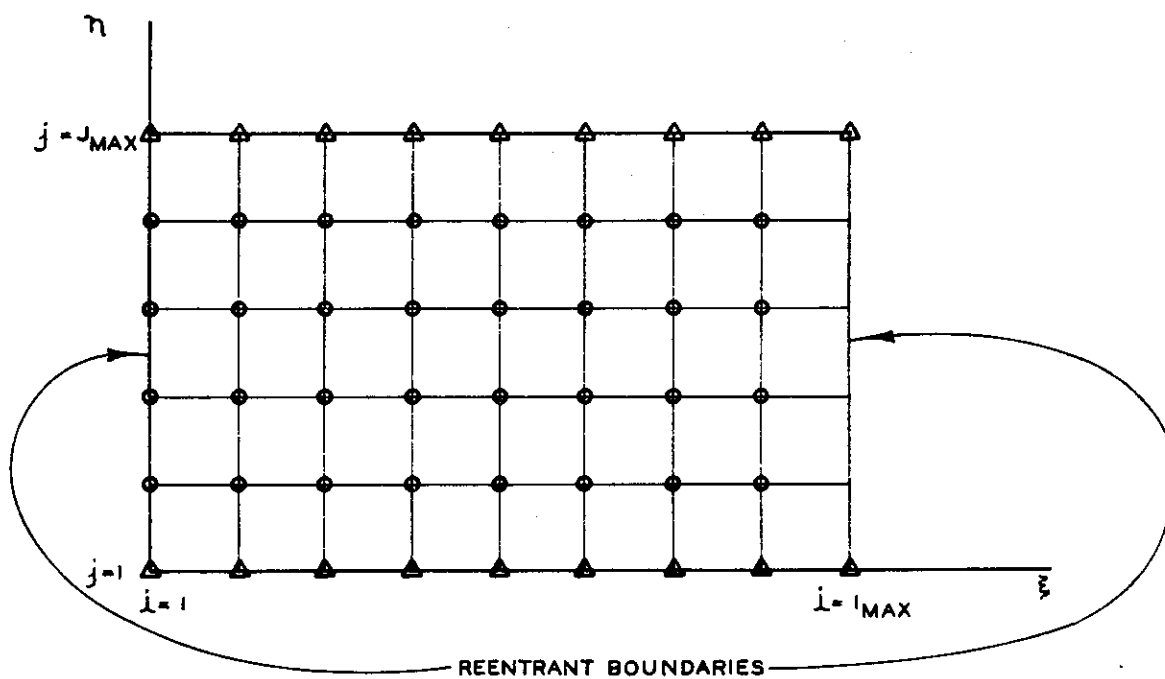


Figure 5. Computational grid in transformed plane

Some discussion of this technique is presented in Ref. [8].

It might be noted that both orthogonal and conformal transformations are special cases of the generation of boundary-fitted coordinate systems as the solutions of elliptic partial differential systems. In both of these cases the curvilinear coordinates satisfy Laplace's equation with one coordinate constant on each boundary, and the normal derivative of the other coordinate equal to zero on each boundary. A conformal system also requires a certain relation between the range of the two curvilinear coordinates.

The same procedure may be extended to regions that are more than doubly connected, i.e., have more than two closed boundaries, or equivalently, more than one body within a single outer body. A river reach containing more than one island would be an example. One such transformation for such a problem is illustrated in Fig. 6.

#### Types of Boundary-Fitted Coordinate Systems

The above discussion of the generation of boundary-fitted coordinates has centered around the idea of using branch cuts to reduce multiply connected regions to simply connected ones in the transformed plane. An example using branch cuts is sketched in Fig. 7. Here the body in the field transforms to the entire bottom boundary of the transformed plane, while the entire surrounding boundary, 1 - 2 - 3 - 4 - 5 - 6, transforms to the top boundary of the transformed plane. The sides of the transformed plane are reentrant boundaries, corresponding to the cut, 8 - 1 and 7 - 6, in the physical field. Thus, in the difference equations, points lying just to the right of the right boundary are identical with corresponding points just to the right of the left boundary. This is



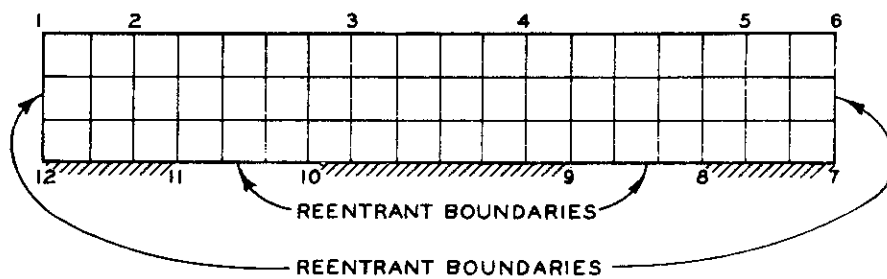
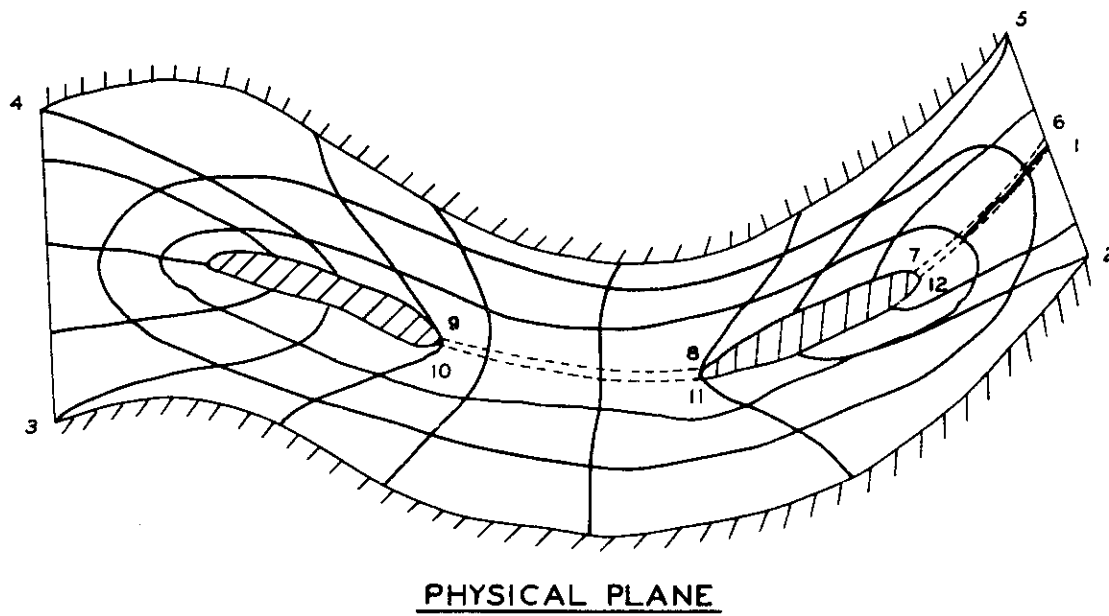
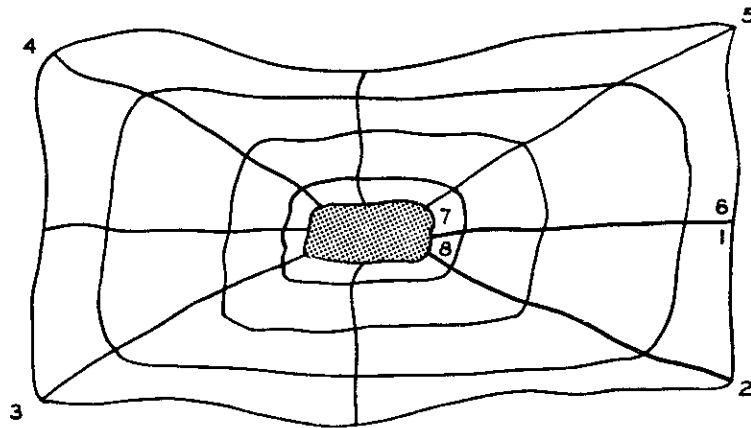
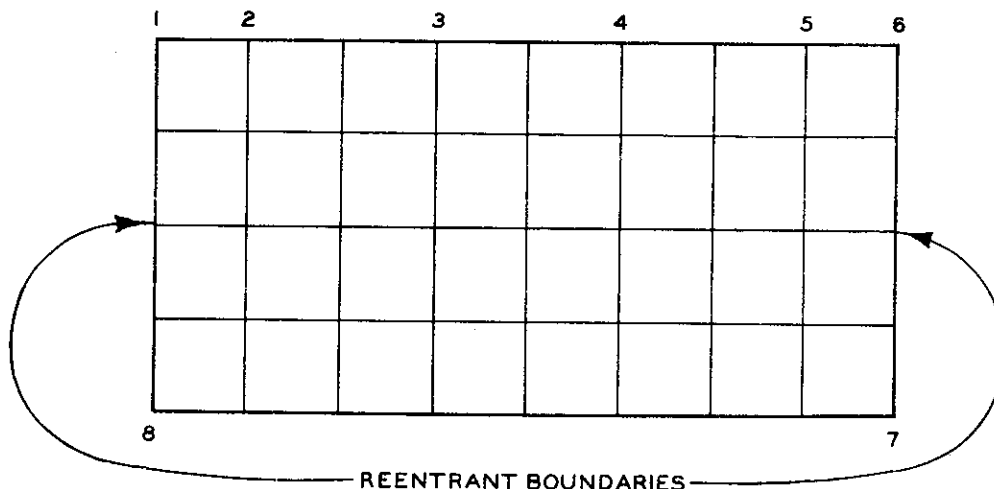


Figure 6. Boundary-fitted coordinates for a river containing two islands



PHYSICAL PLANE



TRANSFORMED PLANE

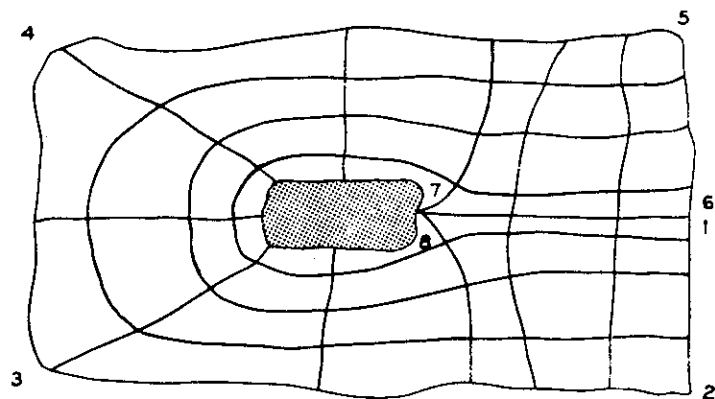
Figure 7. Example of coordinates generated using a branch cut. Placement of body is such that sides are reentrant boundaries

the same type of circumstance that occurs with the familiar cylindrical coordinate system, where  $\theta = 361^\circ$  is the same point as  $\theta = 1^\circ$ . Similarly, points just outside the left boundary are coincident with points just inside the right boundary.

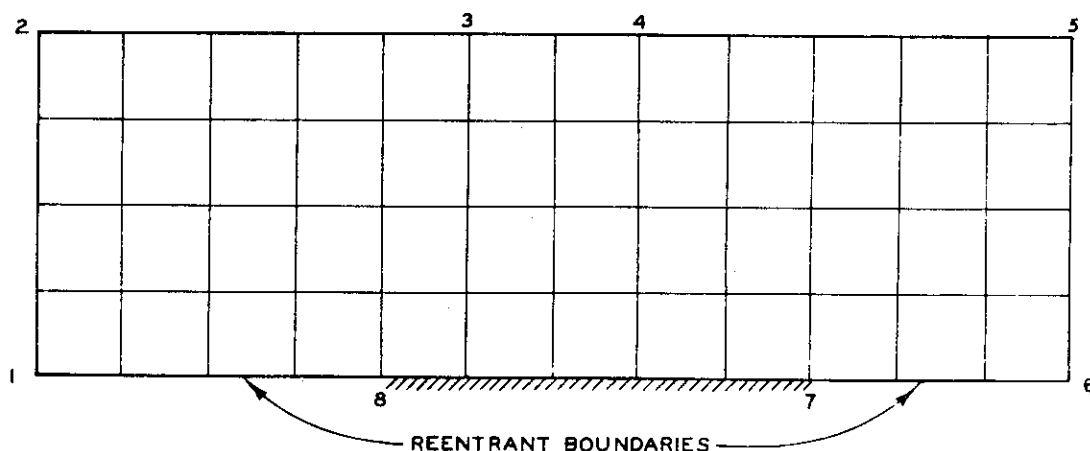
Many variations of this type of coordinate system can be produced, cf. Ref. [3]. For instance, the transformed plane corresponding to the same physical field shown in Fig. 7 can be rearranged as shown in Fig. 8. Now the reentrant boundary, corresponding to the cut, is located on a portion of the bottom of the transformed plane. The coordinate lines that result from these two types of arrangements of the transformed plane are shown on each of the figures. As with all the boundary-fitted coordinate systems, the grid is square in the transformed plane regardless of the line configuration in the physical plane.

Multiple-body fields can also be transformed to simply connected regions, an example of which is shown in Fig. 9. Again there are many different possible arrangements of the transformed plane, all of which are created by sliding the boundary segments around the rectangular boundary of the transformed plane. A number of examples are given in Ref. [3] and Ref. [8].

The other type of coordinate system transformation available leaves the multiplicity of the region unchanged. In this case, bodies in the interior of the physical field are transformed to rectangular slabs or even slits in the transformed plane. Three different possibilities are shown in Fig. 10 for the physical plane shown in Fig. 7. In the case of slits, the physical coordinates and solution variables in general have different values at points on the two sides of the slit, even though such

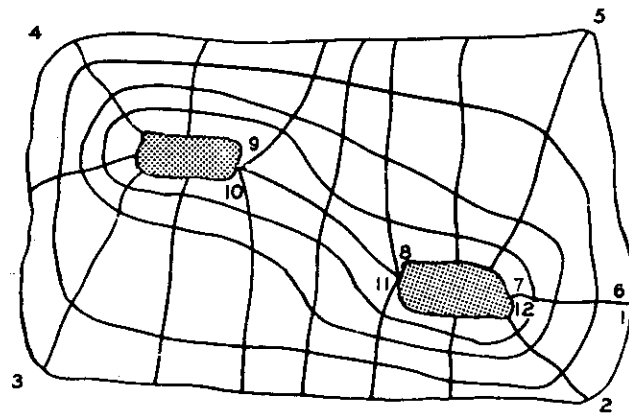


PHYSICAL PLANE

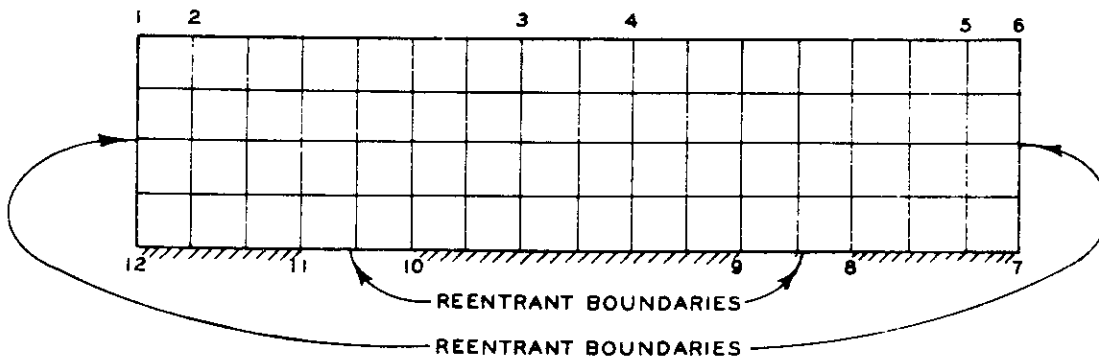


TRANSFORMED PLANE

Figure 8. Example of coordinates generated using a branch cut. Placement of body is such that reentrant boundaries lie on bottom line of the transformed plane

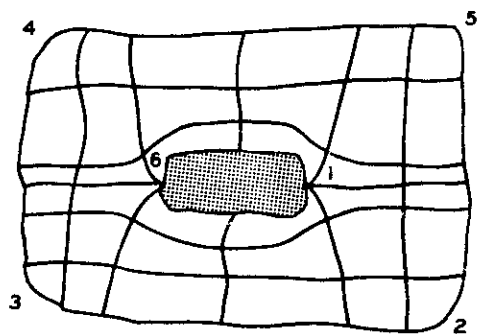


PHYSICAL PLANE

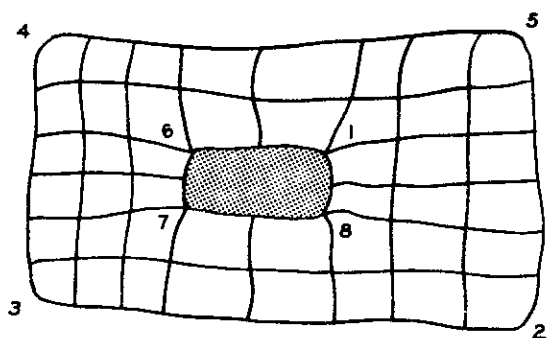
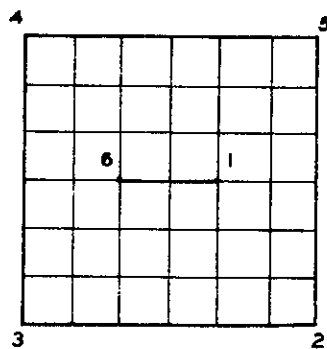


TRANSFORMED PLANE

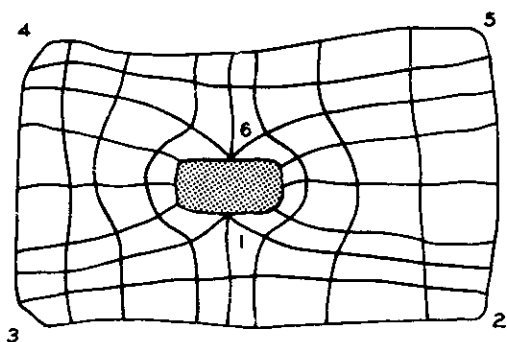
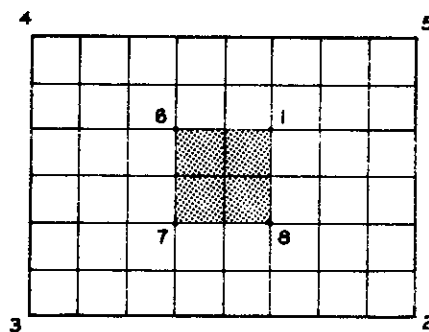
Figure 9. Coordinates generated for a multiple-body field



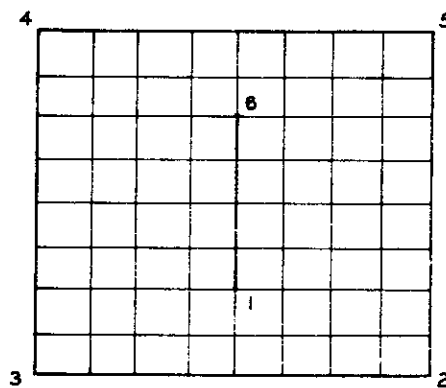
a.



c.



b.



PHYSICAL PLANE

TRANSFORMED PLANE

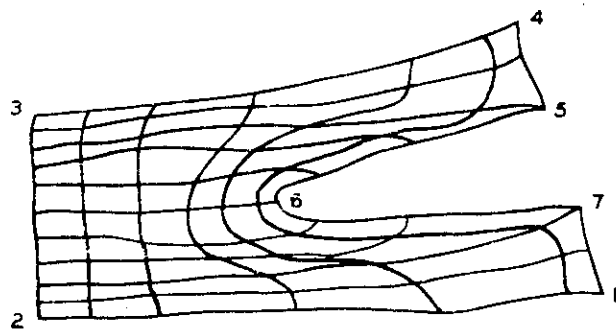
Figure 10. Examples of coordinates generated using slabs/slits

points are coincident in the transformed plane. This does not introduce any approximations, but simply adds a little more bookkeeping to the code. Fields with more than one body in the interior simply result in a like number of slabs and/or slits in the transformed plane.

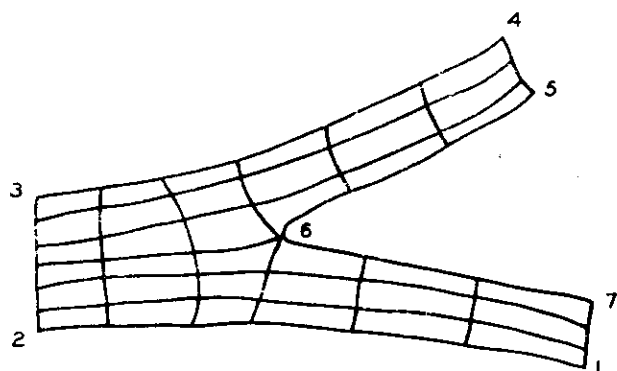
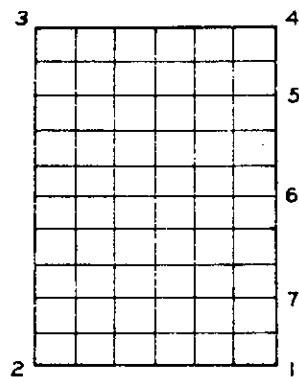
Comparison of all of the above figures shows that different types of transformation may be more appropriate for different physical configurations. A further example of this is the configuration in Fig. 11, shown with three variations. Generally, the slit/slab form is more appropriate for channel-like physical configurations having bodies in the interior, while the other form works particularly well for "unbounded" regions involving external flow about bodies and for regions having an outer boundary that forms a continuous circuit without pronounced corners around the field. The slab is generally superior to the slit unless the boundary has a sharp point. The case of a single channel without any interior bodies is the same in either form. An example of a river reach containing two islands, using horizontal slits rather than the branch cuts previously presented in Fig. 6, is given in Fig. 12.

#### Data Required for Generation of Boundary-Fitted Coordinates

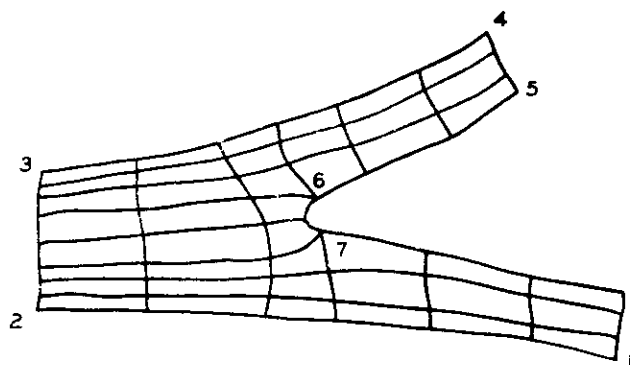
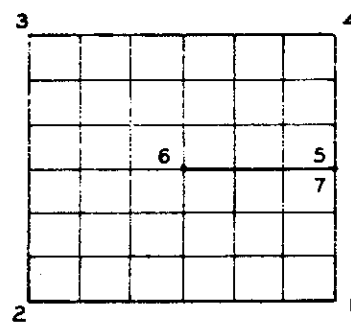
The basic input or data required to generate a boundary-fitted coordinate system are the physical coordinates of points on the boundaries. For example, with reference to Fig. 7, the coordinates of points on the body from 8 around to 7 would be required, with these points being spaced in any manner desired as long as there is a continuous progression from 8 to 7. Similarly, the (x,y) values for points on the outer boundary from 1 to 2, etc., on around to 6 would be required. Again these points



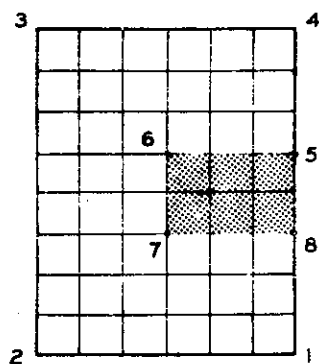
a.



b.



c.

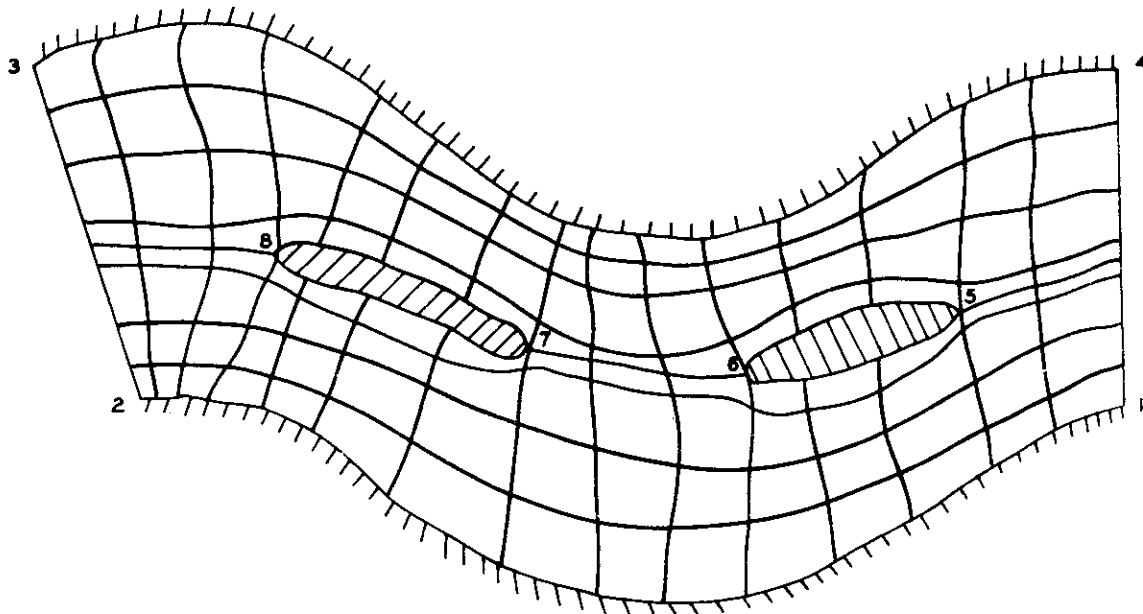


### PHYSICAL PLANE

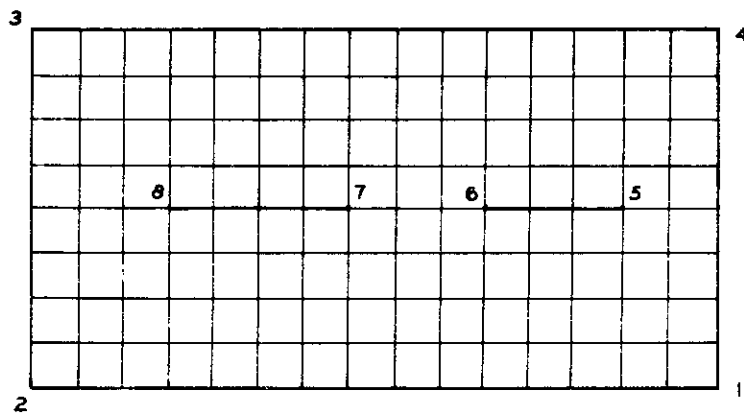
### TRANSFORMED PLANE

Figure 11. Comparison of coordinates without interior body and with slit/slab interior body





PHYSICAL PLANE



TRANSFORMED PLANE

Figure 12. Coordinates generated with slits for a river with two islands

may be spaced around the boundary as desired, with no restriction as to how many points lie on each boundary segment, e.g., between 1 and 2 or between 4 and 5, provided that only the total number of points from 1 around to 6 is the same as from 8 to 7. The coordinates of points must be specified on the entirety of these lines. The coordinates of points on reentrant segments of the boundary in the transformed plane, e.g., 1 to 8 and 6 to 7, are not specified but are free to be determined by the solution.

Similarly, with reference to Fig. 10a, the coordinates of outer boundary points are required in the slab/slit transformations. In addition, body points from 6 to 1 on the lower half of the body and from 1 to 6 on the top half are required. No calculations would be made on the slab sides of Figure 10c or slits of Figures 10a and 10b since values at such points are fixed. Points in the interior of a slab are irrelevant. As always, points may be spaced as desired around the bodies and outer boundary segments.

#### Computer Time Required for Generation of Boundary-Fitted Coordinates

Ref. [ 8 ] indicates that the typical time required to generate a one-body coordinate system without coordinate system control (the functions P and Q are set to zero) is about 2 min on a UNIVAC 1106 computer for a 70 x 30 field (70 points on the body). If P and Q are not zero, so that the spacing of coordinate lines is controlled, the computation time increases. Multiple-body coordinate systems typically require about 6 min for a 70 x 40 field. If these same computations were to be made on a CDC-7600 computer, the times quoted above would be reduced by perhaps

an order of magnitude or more. Therefore, the cost of generating boundary-fitted coordinate systems for use in numerical models will be generally insignificant.

#### COORDINATE SYSTEM CONTROL

Control of the coordinate line spacing in the field can be exercised through the non-zero values given to the Laplacian of the curvilinear coordinates as in Eq. (1), as noted above. With a zero Laplacian, the lines tend to be closely spaced near convex segments and more widely spaced near concave segments. A negative value of the Laplacian causes the lines to move toward lower values of the curvilinear coordinate.

#### Attraction to Other Coordinate Lines and/or Points

This effect is utilized as in Ref. [ 8 ] to achieve attraction of coordinate lines to other coordinate lines and/or points by taking the form of the control functions to be

$$\begin{aligned}
 P(\xi, \eta) = & - \sum_{i=1}^n a_i \operatorname{sign}(\xi - \xi_i) \exp(-c_i |\xi - \xi_i|) \\
 & - \sum_{i=1}^m b_i \operatorname{sign}(\xi - \xi_i) \exp\{-d_i [(\xi - \xi_i)^2 + (\eta - \eta_i)^2]^{\frac{1}{2}}\}
 \end{aligned}
 \tag{5}$$

and an analogous form for  $Q(\xi, \eta)$  with  $\xi$  and  $\eta$  interchanged. The effects of such control is illustrated in Refs. [ 7 ] and [ 8 ]. The efficacy of control to improve the accuracy of a physical solution done on the coordinate system has been noted.

In the  $P$  function, the effect of the amplitude,  $a_i$ , is to attract  $\xi$ -coordinate lines toward the  $\xi_i$ -line, while the effect of the amplitude

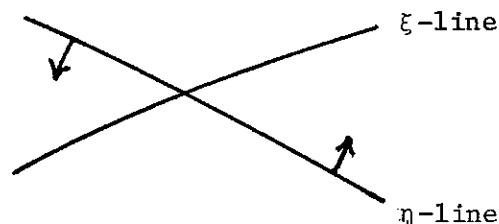
$b_i$  is to attract  $\xi$ -lines toward the single point  $(\xi_i, \eta_i)$ . Note that this attraction to a point is actually attraction of  $\xi$ -lines to a point on another  $\xi$ -line, and, as such, acts normal to the  $\xi$ -line through the point. There is no attraction of  $\eta$ -lines to this point via the  $P$  function. In each case the range of the attraction effect is determined by the decay factors,  $c_i$  and  $d_i$ . With the inclusion of the sign changing function, the attraction occurs on both sides of the  $\xi$ -line, or the  $(\xi_i, \eta_i)$  point, as the case may be. Without this function, attraction occurs only on the side toward increasing  $\xi$ , with repulsion occurring on the other side. A negative amplitude simply reverses all of the above-described effects, i.e., attraction becomes repulsion and vice versa. The effect of the  $Q$  function of  $\eta$ -lines follows analogously. It should be noted that  $P$  and  $Q$  are discontinuous because of the sign function and are equal to sums of second derivatives. As a consequence, the coordinates have continuous first derivatives but discontinuous second derivatives at controlled locations.

In the case of a boundary that is an  $\eta$ -line, positive amplitudes in the  $Q$  function will cause  $\eta$ -lines off the boundary to move closer to the boundary, assuming that  $\eta$  increases off the boundary. The effect of the  $P$  function will be to alter the angle at which the  $\xi$ -lines intersect the boundary, since the points on the boundary are fixed, with the  $\xi$ -lines tending to lean in the direction of decreasing  $\xi$ . If the boundary is such that  $\eta$  decreases off the boundary, then the amplitudes in the  $Q$  function must be negative to achieve attraction to the boundary. In any case, the amplitudes  $a_i$  cause the effects to occur all along the boundary, while the effects of the amplitudes  $b_i$  occur only near selected points on the boundary.

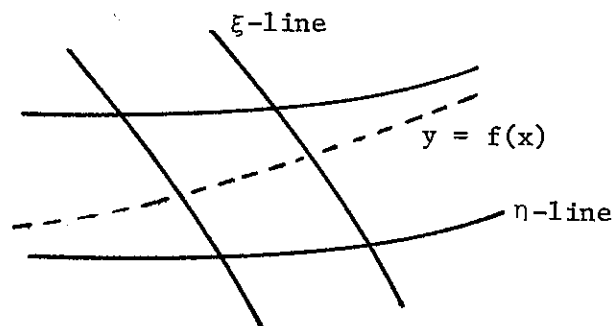
### Attraction to Space Curves and/or Points

If the attraction line and/or the attraction points are in the field, rather than on a boundary, then the attraction is not to a fixed line or point in space, since the attraction line or points are themselves solutions of the system of equations, the functions  $P$  and  $Q$  being functions of the variables  $\xi$  and  $\eta$ . It is, of course, also possible to take these control functions as functions of  $x$  and  $y$ , instead of  $\xi$  and  $\eta$ , and achieve attraction to fixed lines and/or points in the physical field. This case becomes somewhat more complicated, since it must be ensured that coordinate lines are not attracted parallel to themselves. The following development was given in Ref. [9].

Recall that in the above discussion,  $\eta$ -lines are attracted to other  $\eta$ -lines, and  $\xi$ -lines are attracted to other  $\xi$ -lines. It is unreasonable, of course, to attempt to attract  $\eta$ -lines to  $\xi$ -lines, since that would have the effect of collapsing the coordinate system:



When, however, the attraction is to be to certain fixed lines in  $x$ - $y$  space, defined by curves  $y = f(x)$ , care must be exercised to avoid attempting to attract  $\eta$ - or  $\xi$ -lines to specified curves that cut the  $\eta$ - or  $\xi$ -lines at large angles. Thus, in the figure below,



it is unreasonable to attract  $\xi$ -lines to the curve  $f(x)$ , while it is natural to attract the  $\eta$ -lines to  $f(x)$ .

However, in the general situation, the specified line  $f(x)$  will not necessarily be aligned with either a  $\xi$ - or  $\eta$ -line along its entire length. Since it is unreasonable to attract a line tangentially to itself, some provision is necessary to decrease the attraction to zero as the angle between the coordinate line and the given line  $f(x)$  goes to  $90^\circ$ . This can be accomplished by multiplying the attraction function by the cosine of the angle between the coordinate line and the line  $f(x)$ . It is also necessary to change the sign on the attraction function on either side of the line  $f(x)$ . This can be done by multiplying by the sine of the angle between the line  $f(x)$  and the vector to the point on coordinate line.

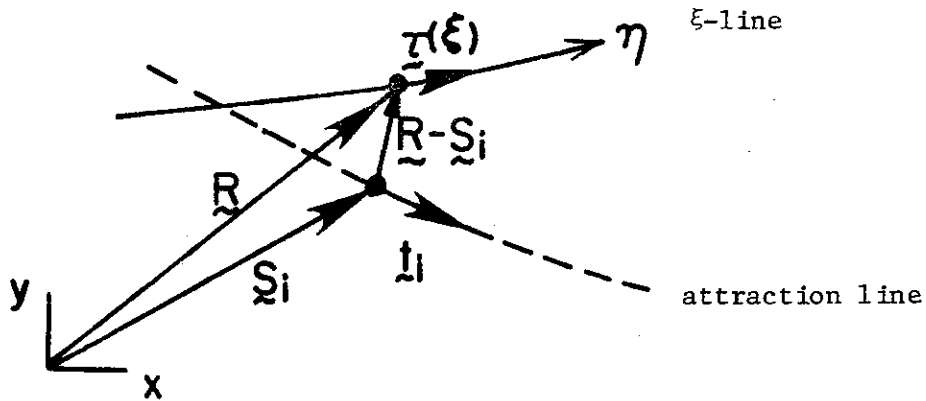
These two purposes can be accomplished as follows. Let a general point on the  $\xi$ -line be located by the vector  $\tilde{R}(x, y)$ , and let the attraction line  $y = f(x)$  be specified by the collection of points  $\tilde{S}(x_i, y_i)$ ,  $i = 1, 2, \dots, n$ . Let the unit tangent to the attraction line be  $\tilde{t}(x_i, y_i)$ , and the unit tangent to a  $\xi$ -line be  $\tilde{\tau}^{(\xi)}$ .

The control functions  $P(x,y)$  and  $Q(x,y)$  may then be logically taken as

$$P(x,y) = - \sum_{i=1}^n a_i(t_i \cdot \tau(\xi)) \frac{[t_i \times (R - S_i)] \cdot k}{|R - S_i|} \exp(-d_i |R - S_i|) \quad (6)$$

$$Q(x,y) = - \sum_{i=1}^n a_i(t_i \cdot \tau(\eta)) \frac{[t_i \times (R - S_i)] \cdot k}{|R - S_i|} \exp(-d_i |R - S_i|)$$

where  $k$  is the unit vector normal to the two-dimensional plane. These relations are evident from the figure below:



Here the term  $t_i \cdot \tau(\xi)$  serves to decrease the attraction to zero as the angle between the  $\xi$ -line and the attraction line approaches  $90^\circ$ . The cross product term changes the sign of the control function on either side of the attraction line to produce attraction on both sides of the line. Again the strength and range of the attraction are determined by the amplitude,  $a_i$ , and the decay factor,  $d_i$ , respectively.

These functions depend on  $x$  and  $y$  through both  $R$  and  $\tau^{(\xi)}$  or  $\tau^{(\eta)}$ , and thus must be recalculated at each point as the iterative solution proceeds. This form of coordinate control will therefore be more expensive than that based on attraction to other coordinate lines.

There is no real distinction between "line" and "point" attraction with this type of attraction. "Line" attraction here is simply attraction to a group of points that form a line  $f(x)$ . If line attraction is specified, then the tangent to the line  $f(x)$  is computed from the adjacent points on the line. If point attraction is specified, then the "tangent" must be input for each point. The tangents to the coordinate lines are computed from the relations given in Ref. [3].

#### Control Functions from Boundary-Point Distributions

With the Laplacians of the coordinates equal to zero, the line spacing in the field will not be greatly affected by the distribution of the boundary points, except very near the boundaries. In fact, if the control functions are not consistent with the boundary point distribution, very large changes in the metric coefficients will occur near the boundaries. Values of the control functions may be determined from the 1D boundary point distribution such that the line spacing in the field will generally follow that on the boundary. This concept was introduced in Ref. [10] and is discussed in Ref. [7] as generalized to 3D in Ref. [11]. However, in the use of control functions that are 1D, it should be noted that excessive concentration of lines can occur near sharp convex corners as discussed in Ref. [7].

With Eq. (3) evaluated in 1D on a straight  $\eta$ -line coincident with the  $x$ -axis, we have, since  $x_\eta = y_\xi = 0$  in this case,



$$\alpha x_{\xi\xi} = -\alpha P(\xi) x_{\xi} \quad (7)$$

The reason for the choice of the form of the control functions in Eq. 3 becomes clear, since  $\alpha$  cancels from this equation to leave

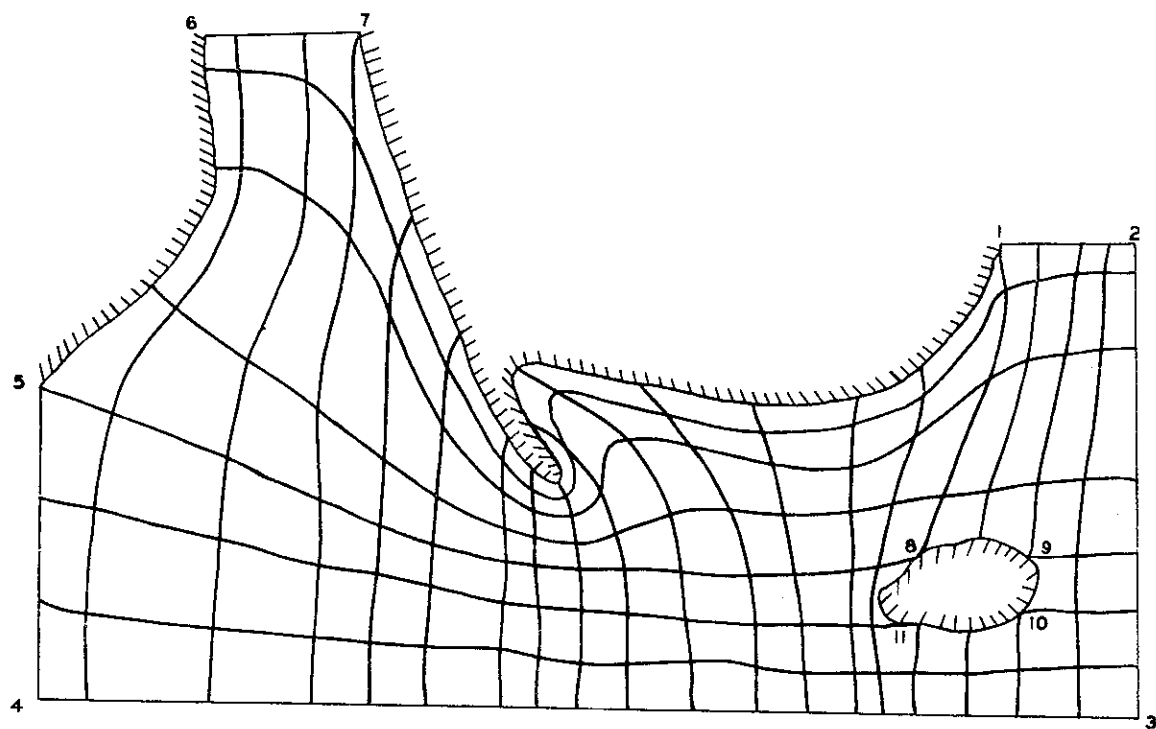
$$P(\xi) = -x_{\xi\xi}/x_{\xi} \quad (8)$$

Thus the control function,  $P(\xi)$ , can be determined from the specified boundary point distribution,  $x(\xi)$ . Generalizing,  $x$  is replaced by arc length along the  $\xi$ -line, and the effect will be qualitatively the same when this line is curved. (Compare Ref. [7] for more detail.)

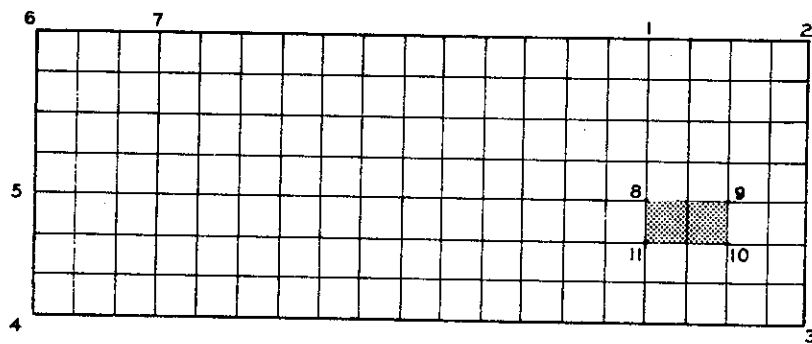
If this value of the control function is then used throughout the field, the  $\xi$ -line distribution in the field will generally follow the specified distribution of the end points of these lines on the boundary. With different point distributions on two boundaries, values of the control function  $P(\xi, \eta)$  in the field between can be determined by 1D interpolation in  $\eta$  between the values determined in the above manner on the two  $\eta$ -line boundaries. An analogous development applies for the determination of the control function  $Q(\xi, \eta)$  from interpolation in between 1D evaluations on two  $\xi$ -line boundaries. This interpolation was introduced in Ref. [12] in a 2D coordinate system.

#### SYSTEM CONFIGURATION

In the present model, the physical field may have both external and internal boundaries of arbitrary shape. The field in the transformed plane is rectangular with rectangular holes corresponding to any internal boundaries. This configuration is illustrated in Fig. 13. Boundary



PHYSICAL PLANE



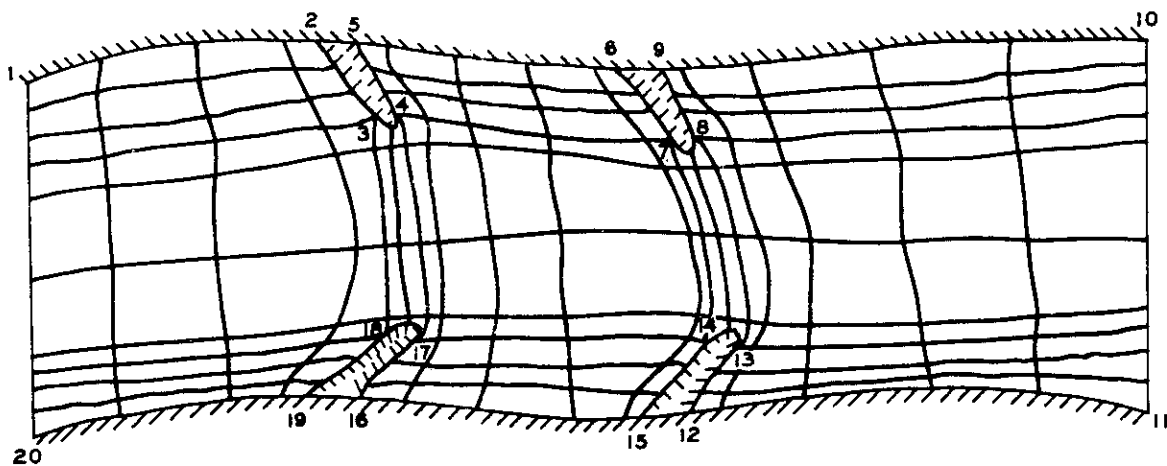
TRANSFORMED PLANE

Figure 13. Example of coordinates generated in a field containing a jetty and an island

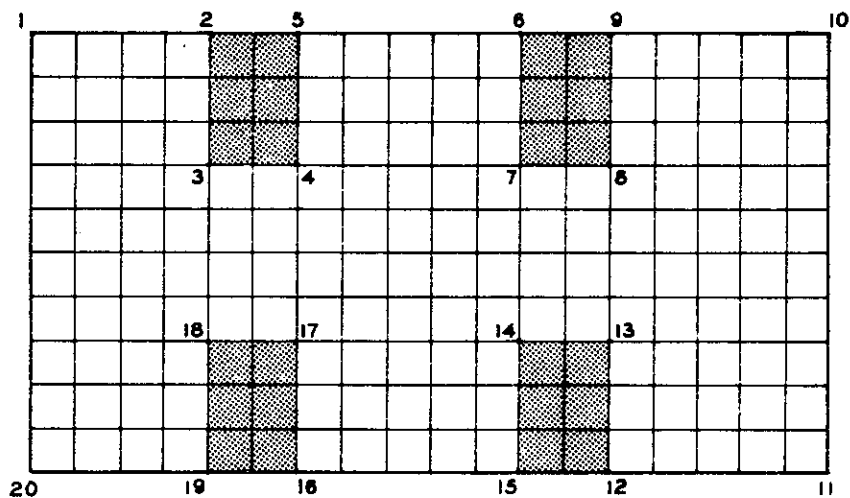
intrusions may be transformed either to portions of the rectangular outer boundary of the transformed region, as in Fig. 13, or to slabs protruding inward from this boundary as in Fig. 14. A general discussion of possible configurations is given in Ref. [3]. Various outlet shapes and locations, as well as internal obstacles and boundary protrusions such as weirs, can be treated by the same code with only changes in the input. This input consists of the physical cartesian coordinates of the points selected on each segment of the physical boundaries. A small front-end code was written to provide certain line segments (linear, quadratic, and cubic polynomials) with linear or exponential distributions thereon automatically.

The code automatically calculates control functions  $P(\xi, \eta)$  and  $Q(\xi, \eta)$  for the coordinate generation equations (3) from the boundary point distribution as discussed above. These functions are calculated from the 1D relations on each boundary segment and are interpolated linearly into the field between opposing boundary sections in the transformed plane.

In addition, attraction of coordinate lines to other coordinate lines and/or points, and to specified lines and/or points in space, also discussed above, is provided through input quantities. This input consists of the coordinate lines and/or points, and the specified space curves and/or points, to which the attraction is to be made and the amplitudes and decay factors for the corresponding attractions.



PHYSICAL PLANE



TRANSFORMED PLANE

Figure 14. Boundary-fitted coordinates for a river containing dikes

Several examples of coordinate systems produced by this code are given in Figs. 15-19. Examples of applications of such systems appear in Ref. [4]-[6]. Two further examples, together with complete input listings for the code, follow the description of the code in Part B.

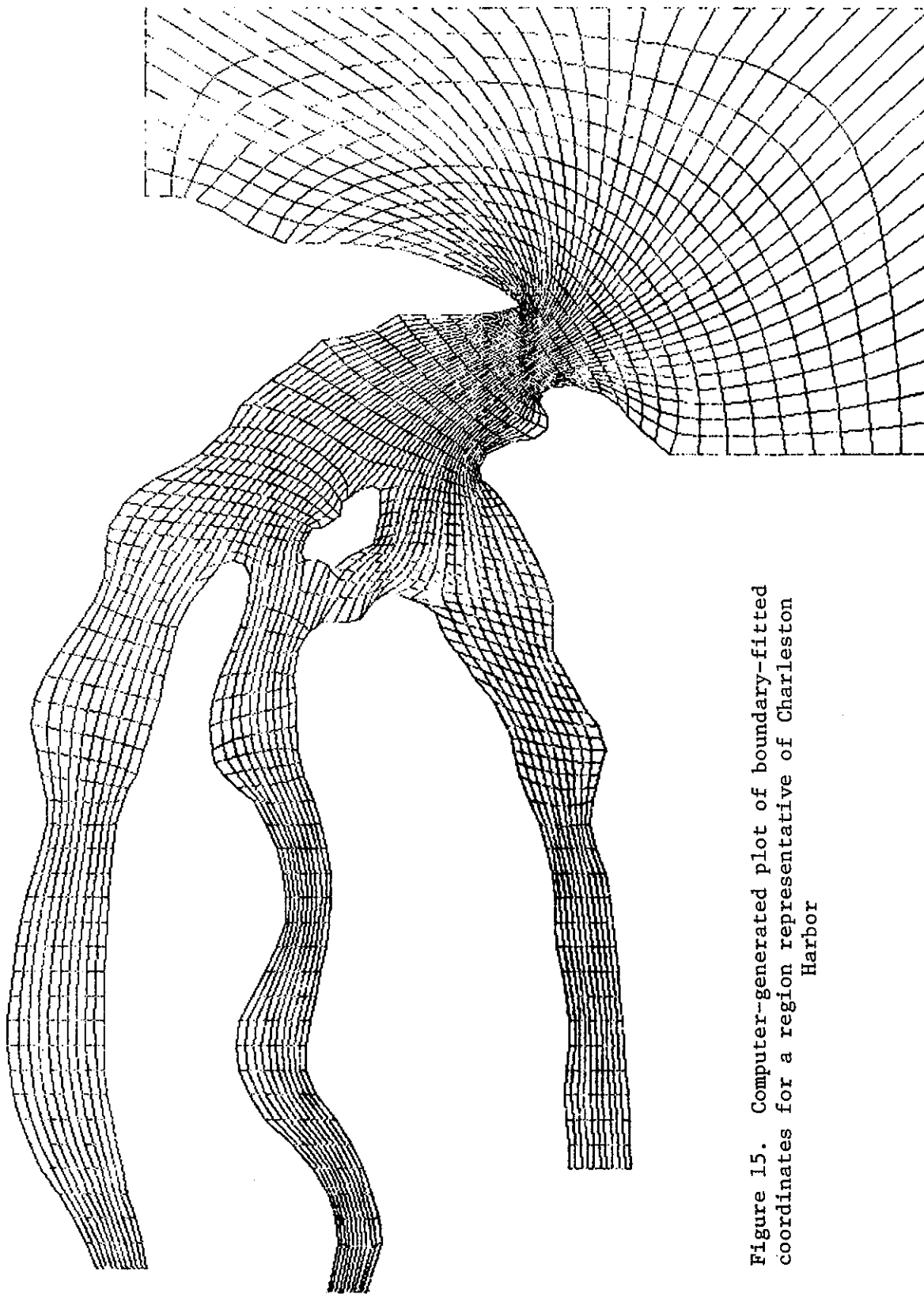


Figure 15. Computer-generated plot of boundary-fitted coordinates for a region representative of Charleston Harbor

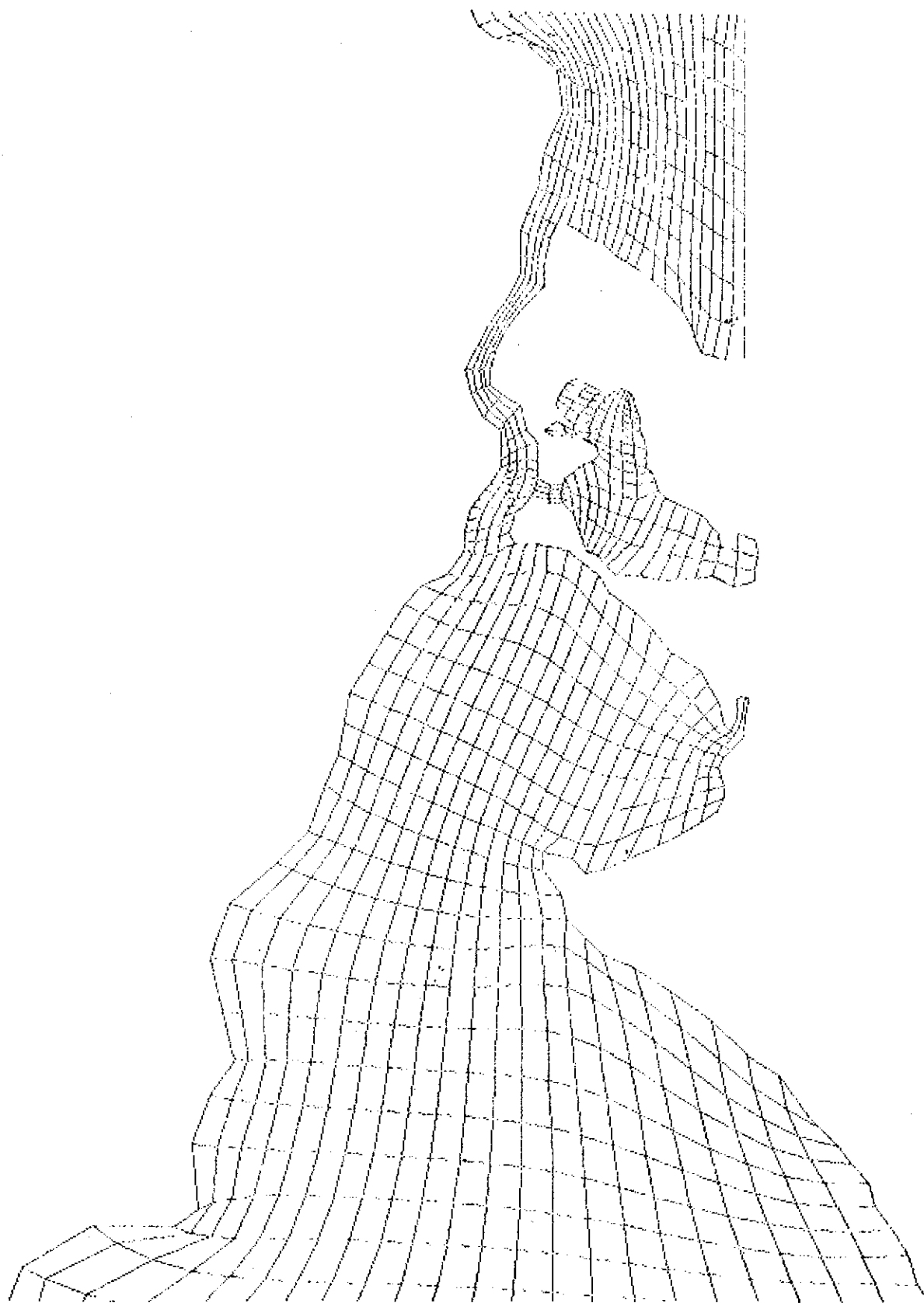


Figure 16. Coordinate system for a portion of Lake Pontchartrain

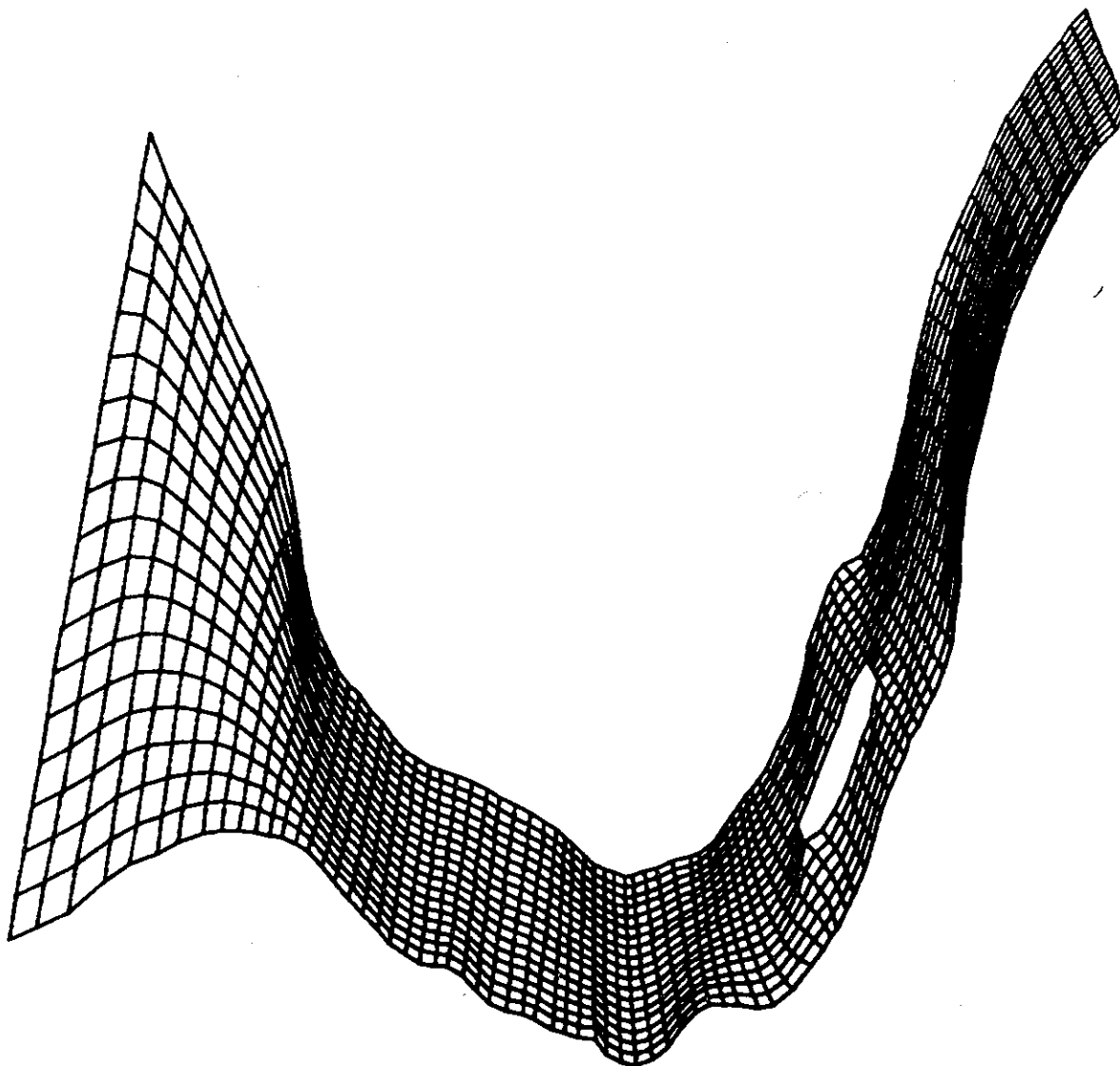


Figure 17. Hypothetical estuary similar to Delaware River (from B. H. Johnson, Waterways Experiment Station, Vicksburg)



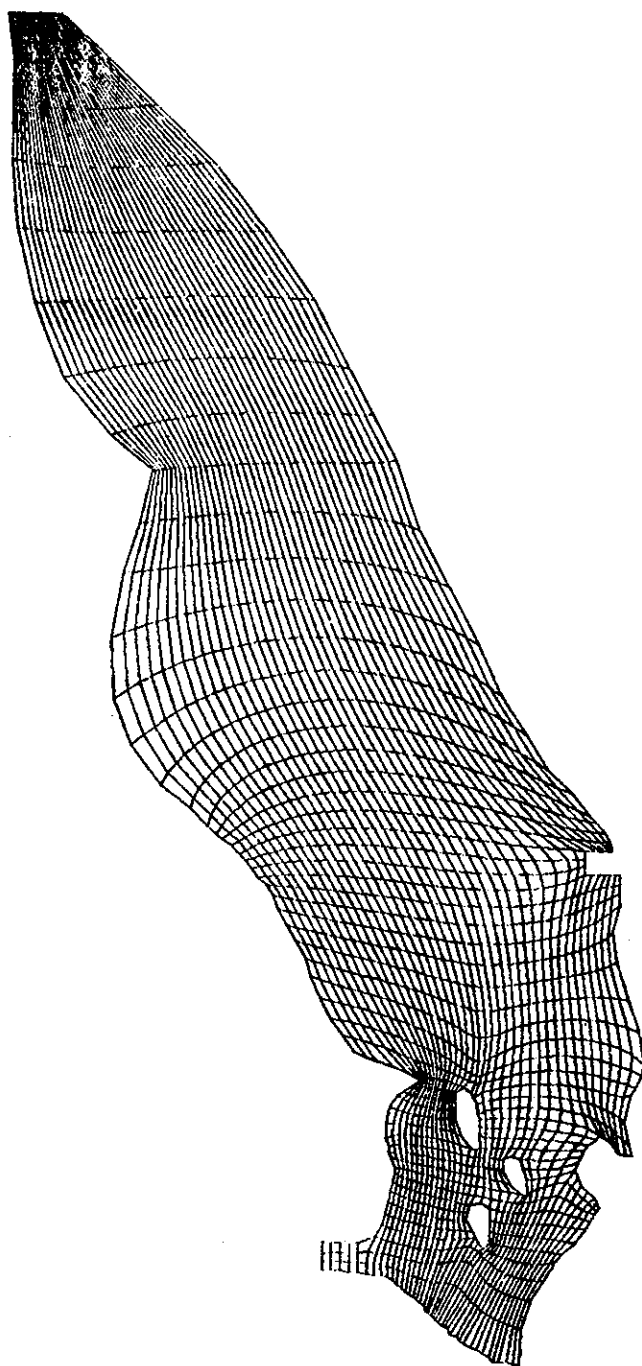


Figure 18. Coordinate system fitted to Lake Erie

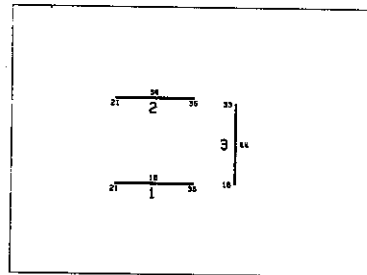
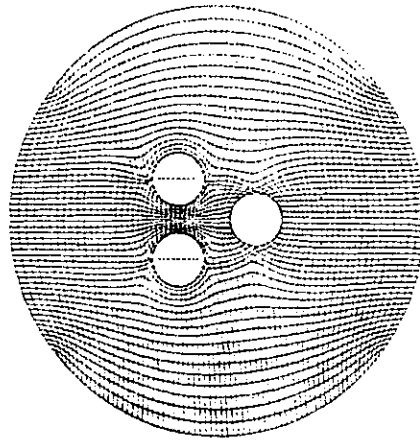


Figure 19. Transformation to slits

## REFERENCES

1. Thompson, J. F., Zahir U. A. Warsi, and C. Wayne Mastin. "Boundary-Fitted Coordinate Systems for Numerical Solution of Partial Differential Equations - A Review," Journal of Computational Physics, to appear in mid-1982.
2. Thompson, J. F. (ed.). Numerical Grid Generation, Elsevier (1982).
3. Thompson, J. F. "General Curvilinear Coordinate Systems," in Ref. [2].
4. Thompson, J. F. "Numerical Modeling of 2D Width-Averaged Flows Using Boundary-Fitted Coordinate Systems, with Application to Selective Withdrawal from Reservoirs," MSSU-EIRS-ASE-82- , Mississippi State University (1982).
5. Johnson, B. H. and J. F. Thompson. "A Discussion of Boundary-Fitted Coordinate Systems and Their Applicability to the Numerical Modeling of Hydraulic Problems," Misc. Paper H-78-9, U. S. Army Engineer Waterways Experiment Station, Vicksburg, MS (1978).
6. Johnson, B. H. "Numerical Modeling of Estuarine Hydrodynamics on a Boundary-Fitted Coordinate System," in Ref. [2].
7. Thompson, J. F. "Elliptic Grid Generation," in Ref. [2].
8. Thompson, J. F., F. C. Thames, and C. W. Mastin. "TOMCAT - A Code for Numerical Generation of Boundary-Fitted Curvilinear Coordinate Systems on Fields Containing any Number of Arbitrary Two-Dimensional Bodies," Journal of Computational Physics, 24, 274 (1977).
9. Thompson, J. and W. Mastin. "Grid Generation Using Differential Systems Techniques," Numerical Grid Generation Techniques, NASA Conf. Publication 2166, 37 (1980).
10. Warsi, Z. U. A. and J. F. Thompson. "Machine Solutions of Partial Differential Equations in the Numerically Generated Coordinate Systems," MSSU-EIRS-ASE-77-1, Mississippi State University (1976).
11. Thomas, P. D. "Construction of Composite Three Dimensional Grids from Subregion Grids Generated by Elliptic Systems," AIAA Computational Fluid Dynamics Conference, Palo Alto, 24 (1981).
12. Middlecoff, J. F. and P. D. Thomas. "Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations," AIAA 79-1462, AIAA 4th Computational Fluid Dynamics Conference, Williamsburg (1979).

PART B  
COORDINATE CODE

The present code (WESCØR) differs from a previous version (TOMCAT) described in Ref. [8] in that the latter does not provide for slits and/or slabs in the interior of the transformed plane. Also, branch cuts (if used) in the present code are restricted to the entire left and right sides of the outer rectangle in the transformed region. Finally, the present code includes a more extensive means of coordinate line control, involving attraction to space lines/or points and also involving determination from boundary point distributions.

The code for the numerical generation of the boundary-fitted coordinate system from the equations of Part A, together with a front-end code to generate boundary point distributions and a plot code, is discussed below. These codes were implemented on the CRAY-1 computer at the Air Force Weapons Laboratory, Kirtland AFB, New Mexico.

WESCØR (Coordinate System)

This code generates the boundary-fitted coordinate system by solving a set of elliptic partial differential equations by SOR iteration as discussed in Part A. Attraction of coordinate lines to other coordinate lines and/or points, and to specified lines and/or points in space, is included. The shape and configuration of the boundary are arbitrary, except that the outer boundary must be closed. There may be an arbitrary number of internal closed boundaries transforming to either slits or slabs as discussed in Part A.

The input to this code consists of the point distribution on the boundary of the region, several quantities in connection with the control

of the coordinate line spacing, and the parameters associated with the iterative solution process. This input is described in detail below.

The file output from the code LINES can be used directly as a part of the input to this code from file 10. A simplified flow chart of WESCØR is shown in Figure 20.

#### Boundary Configurations

Arrays. The dependent variable field arrays are X and Y, which contain the cartesian coordinates (x,y) for each grid point. The indices (I,J) of these arrays correspond to the curvilinear coordinates ( $\xi, \eta$ ), and run from 1 to IMAX and JMAX, respectively. The increments  $\Delta\xi$  and  $\Delta\eta$  in the difference expressions are thus equal to unity by construction. (These increments cancel from all the difference equations and are thus irrelevant.)

In order to treat slit configurations, for which a closed interior boundary in the physical region is collapsed to a slit in the transformed region, there are four other coordinate arrays, XL, YL and XU, YU, which contain the cartesian coordinates on the two sides of the slit. The first index of these arrays corresponds to the location of the point relative to the left end of horizontal slits, or relative to the lower end of vertical slits, this end index being designated unity. The other index identifies the particular slit. For horizontal slits the coordinates on the lower side are in XL and YL, while those on the upper side are in XU and YU. Vertical slits have the coordinates on the left side in XL and YL, and those on the right side in XU and YU.

There is also a field array LSLIT(I,J) containing the point type for each point. This array identifies each point as being on a slit,

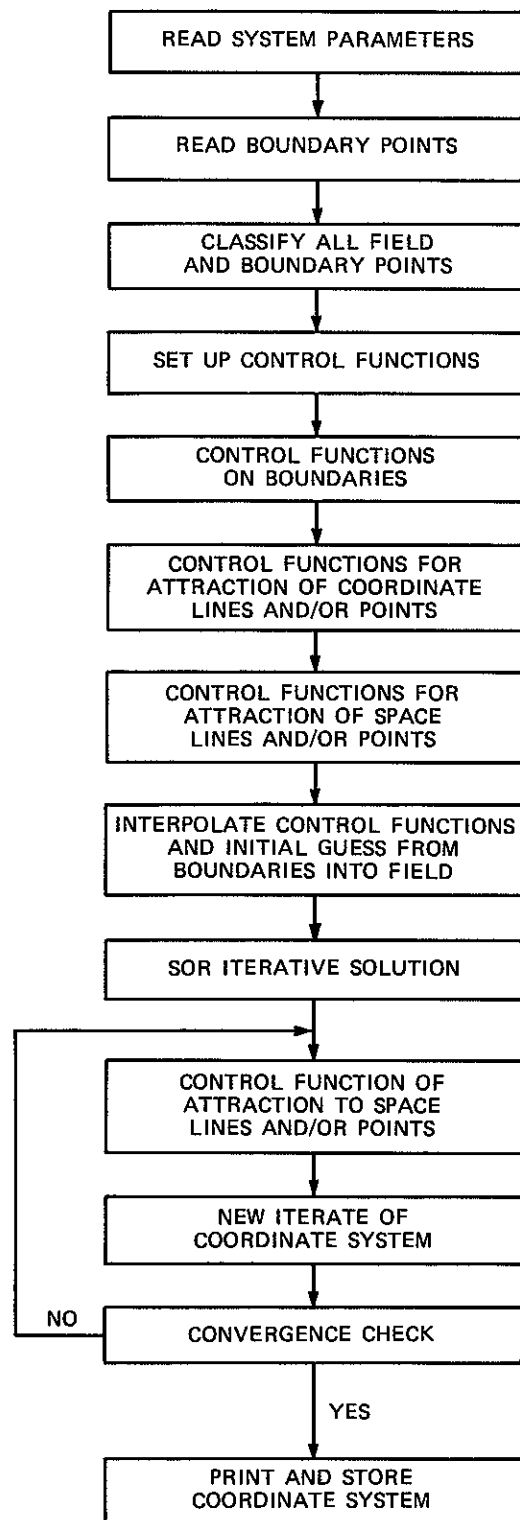
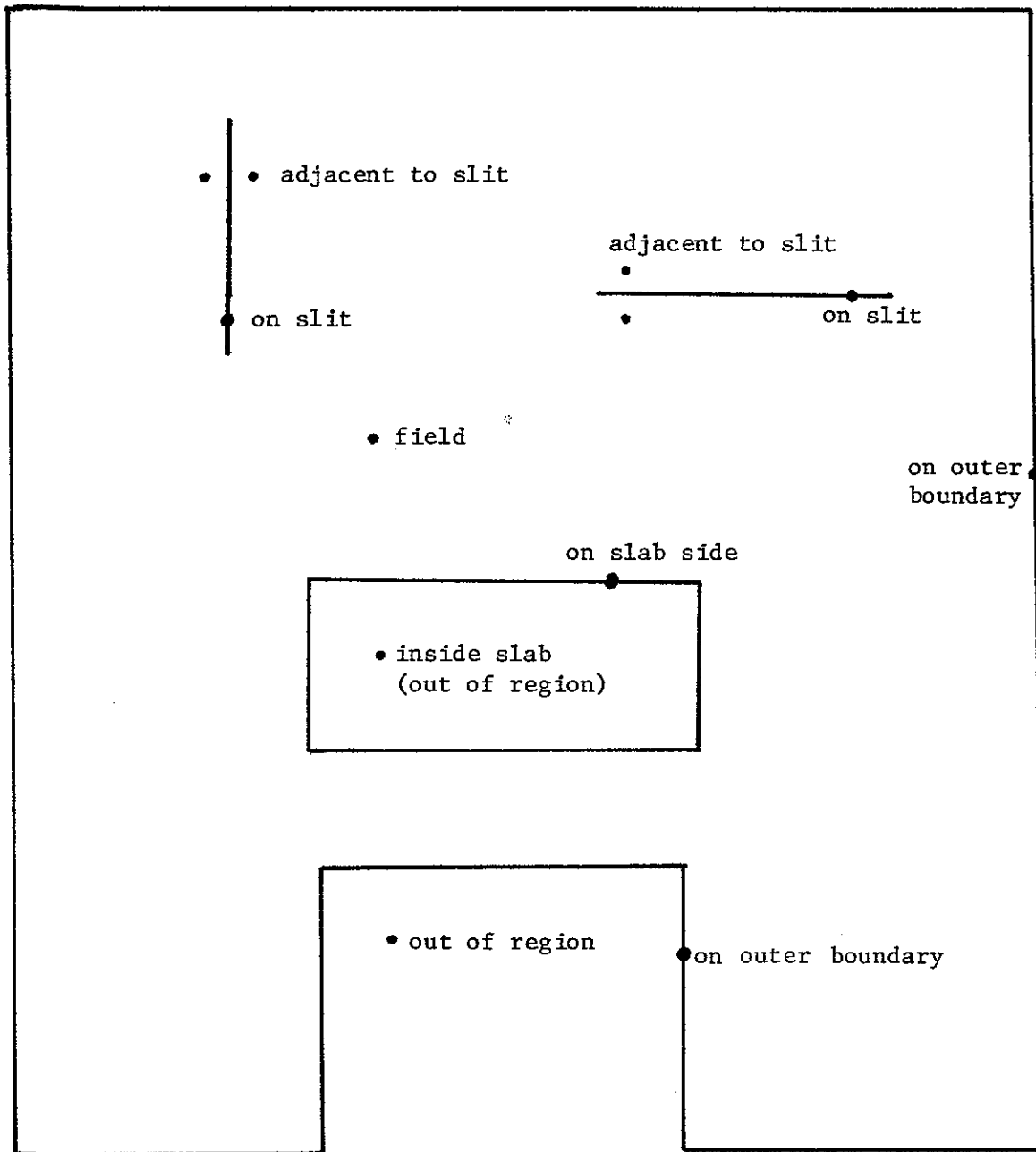


Figure 20. Simplified flow chart

adjacent to a slit, on a slab side, on an outer boundary, in the field, or out of the computation region (inside a slab), as illustrated on the diagram below:

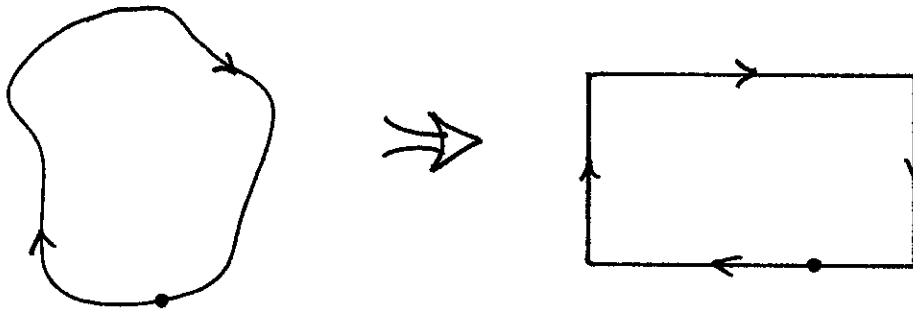


The coordinate system control functions  $P$  and  $Q$  for each point are contained in the field arrays  $RXI(I,J)$  and  $RETA(I,J)$ , respectively. There are also arrays  $RXIL$ ,  $RETAL$  and  $RXIU$ ,  $RETAU$ , analogous to the array  $XL$ , etc., discussed above, which contain the values of these functions on the two sides of the slits. The acceleration parameters for the iteration at each point are in the field array  $WACC(I,J)$ .

Configuration types. The cartesian coordinates of the points on the entire boundary of the physical region, i.e., the closed outer boundary and any internal boundaries, must be input. There are two basic types of overall configuration included in the code. In one the connectivity of the transformed region is the same as that of the physical region, i.e., the closed outer boundary of the physical region corresponds to a closed outer boundary of the transformed region. With the other type, one branch cut is introduced in the physical region so that the closed outer boundary and one inner boundary of the physical region transform to the bottom and top of a rectangle forming the outer boundary of the transformed region. The left and right sides of the transformed region then correspond to the branch cut in the physical region. Points on these sides therefore are not input but rather are calculated as part of the solution.

Rectangular outer boundary. If the outer boundary of the physical region is to correspond to a rectangle forming the outer boundary of the transformed region, then the points on this boundary can be input in clockwise succession around the outer rectangle of the transformed region as in the diagram below. If the outer boundary of the physical region

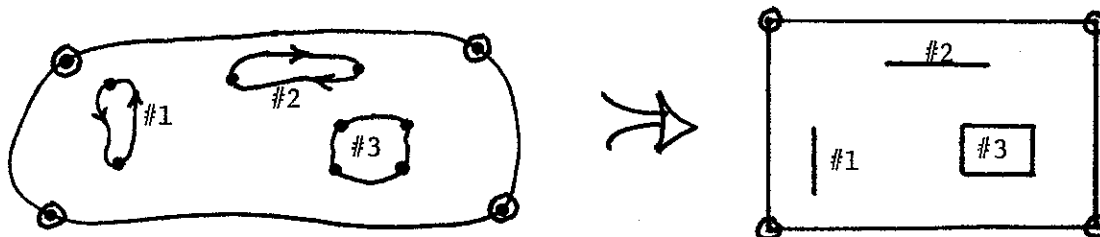




is a circle, then the points on this circle can be generated internally by the code, requiring input only of the radius (YINFIN) and cartesian coordinates of the center (X0INF,Y0INF) of the circle, together with the cartesian coordinates of the angular position (AINFIN) and indices (INFXI,INFETA) of the point at which the clockwise succession of points around the outer rectangle is to start, and the total number of points on the circle (NINF). As above, the points will be placed in clockwise succession around the circle or boundary of the physical region and the rectangular boundary of the transformed region. The treatment of the outer boundary is determined by the input parameter IBNDRY.

An alternative procedure for inputting the outer boundary is to input each straight segment of this boundary of the transformed region as a slab side in the manner described below for internal boundaries.

Internal boundaries (slits/slabs). Internal boundaries in overall configurations of the former type introduced above correspond to either slits or slabs in the transformed region:



In the case of slits, the points are input in clockwise succession beginning at the right end for horizontal slits or counter-clockwise beginning with the top for vertical slits, and are placed in the arrays XL, etc., described above. For slabs, the four sides are input independently and the succession of points may be in either direction on each side. In fact, it is not even necessary for the four sides of one slab to be input in succession; the sides of all slabs in the field may be placed in any order in the input. The coordinates of the points on slab sides are placed directly in the field arrays X and Y. This input of boundary segments corresponding to slits or slabs is accomplished as follows.

For horizontal slits, the  $\xi$ -indices(I) of the left and right ends are placed in the arrays LB1 and LB2, respectively. The  $\eta$ -index (J) of the entire slit or slab side is placed in the array LB3. In the case of vertical slits, the  $\eta$ -indices (J) of the bottom and top go in LB1 and LB2, while the  $\xi$ -index (I) goes in LB3. Slab sides are treated in the same manner except that, since the points thereon may be input in either direction, LB1 and LB2 contain the indices of the end points of the side in either order, i.e., LB1 may exceed LB2. The points are input from LB1 to LB2.

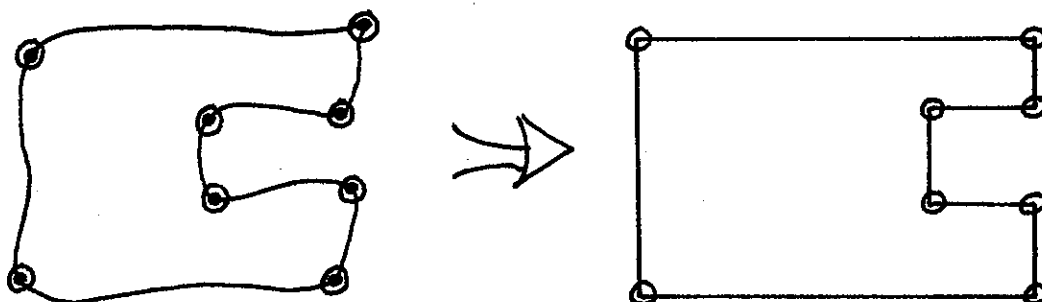
For both slits and slab sides, a flag is placed in an array LTYPE to designate the segment as a slit or slab side in horizontal or vertical orientation:

- +1 horizontal slit
- +2 vertical slit
- 1 horizontal slab side
- 2 vertical slab side

The code computes the number of points on the slit or slab side from the

values of LB1 and LB2 and places this value in the array LPT. All of these arrays are single-dimension arrays, there being one set of parameters for each slit or slab side. The total number of slits and slab sides, including those on the outer boundary as described below, is specified by the input parameter NBDY.

Outer boundary intrusions. As noted above, the outer boundary can be input in segments as slab sides. This is illustrated below.



This is done just as described above for internal boundaries except that values of -11 and -12, respectively, are input for LTYPE for horizontal and vertical segments of the outer boundary.

Branch cut. With the other type of overall configuration, involving a branch cut, the outer boundary and the internal boundary connected to the cut are both input clockwise from the points joined by the cut. As noted above, these points are placed on the top and bottom of the rectangle forming the outer boundary of the transformed region. This type of configuration is elected through the input parameter NREN. Additional internal boundaries can be input as either slits or slabs exactly as described above.

Boundary input. Provision is made for reading the boundary points either from card images (x and y for one point to a card in 2F10.0

format) or from the output of the LINES code described below, as determined by the input parameter ISLIT. The outer boundary must be input as segments of slab sides if this boundary is included on the output of the LINES code.

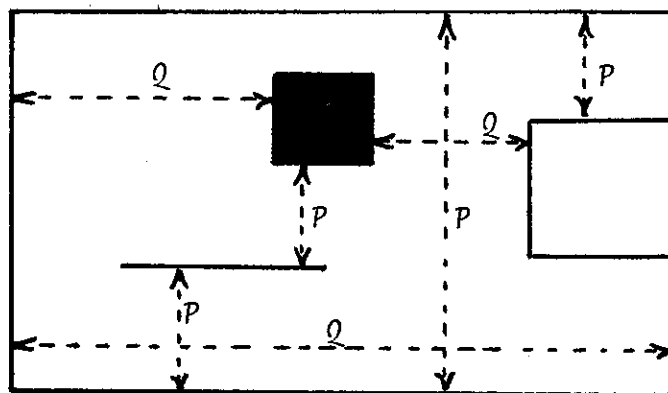
### Control Functions

Coordinate system control is included through both the attraction of coordinate lines to other coordinate lines and/or points and to specified lines and/or points in the physical region, as described in Part A. (For completeness, provision is made for repulsion as well as attraction.)

Attraction to coordinate lines and/or points. The first of these requires the input of the index (indices) of the curvilinear coordinate line, together with the associated attraction amplitude and decay factor, for each line (point) to which the attraction is made. For attraction to lines, the index, amplitude, and decay factor are placed in the arrays JLN, ALN, and DLN, respectively, while for attraction to points, the corresponding arrays IPT, JPT, APT, and DPT are used.

Attraction to space lines and/or points. For attraction to specified lines and/or points in space, the input is similar in regard to the amplitude and decay factors, using the arrays APT and DPT. It is necessary, of course, to also input the cartesian coordinates of the points on the line, or the isolated points, to which the attraction is made. These coordinates are placed in the arrays XPT and YPT. For attraction to points, it is also necessary to input the components of a vector normal to the desired direction of the attraction for each point, these components being placed in the arrays VEC1 and VEC2.

Effect of boundary point distribution. In addition to the above types of attraction, the control functions also include the effect of the boundary point distribution discussed in Part A. This is done by evaluating one of the control functions on each boundary segment in the transformed region ( $P$  on  $\eta$ -lines,  $Q$  on  $\xi$ -lines) from the one-dimensional relations in terms of arc length discussed in Part A. These values are placed in the arrays RXI and RETA, except for slits where the arrays RXIL, etc., are used in the manner described above for XL, etc. Values of the control functions in the field are then interpolated linearly between facing boundary segments,  $P$  being interpolated vertically and  $Q$  horizontally, as illustrated in the following diagram.

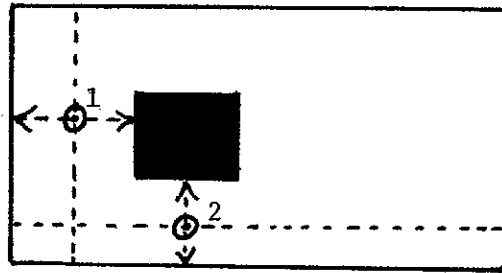


This evaluation is done first and then the contributions to the control functions from the line and point attraction is added to the arrays RXI and RETA in the field.

### Iterative Solution

Initial guess. The initial guess for the values of the cartesian coordinates in the field, i.e., the values in the arrays X and Y in the field, that is used to start the iterative solution is obtained by the

same type of interpolation between facing segments described above for the control functions, except that both X and Y are interpolated between the pair of facing segments with the smallest separation in the transformed region. Thus values at point 1 in the figure below would be obtained by horizontal interpolation, but at 2 the interpolation would be vertical.



Since very strong control functions can sometimes make the convergence of the iterative solution difficult in complicated configurations, provision is made for first converging the field with the control functions set to zero and then re-converging in steps as these functions are increased to full value. Actually this feature is rarely needed.

Acceleration parameters. As discussed in Part A the solution for the cartesian coordinates in the field is done by SOR iteration. Either a uniform value of the acceleration parameter can be input as R(1) or the code will calculate a locally optimum value at each point in the field, these values being placed in the field array WACC. This calculation is discussed in Ref. [8], where it is noted that the values obtained are not truly optimum in all cases. Therefore this provision has not been found to be as generally efficient as simply using a uniform value, since the calculation of the acceleration parameter involves a square root and hence is time-consuming. The uniform value should be around 1.85 for

large fields. This value should be decreased for strong control functions or small fields.

Iterative process. The iteration continues until either the magnitude of the changes in the values of  $x$  and  $y$  at each point in the field between iterations is less than the tolerances input as  $R(2)$  and  $R(3)$ , respectively, or until the maximum number of iterations allowed (input as  $ITER$ ) is reached. In the latter case the partially converged solution is stored on file 10 for restart. The input parameter  $IDISK$  can cause the code to read this partially converged solution from file 10 and continue the iterations. This parameter also controls the disposition of the final solution, which is normally stored on file 11 for use in the flow solution, but can be simply printed without being stored if desired. Various other input parameters, such as print options, etc., are explained in the detailed input instructions given below and in the source listing.

#### Code Operation

Initial input and setup. The  $WESCOR$  code uses the values of  $NDIM$ ,  $NDIM1$ ,  $NDIM2$ , and  $NDIM3$ , which are assigned by a  $DATA$  statement, to determine if the problem specified by the input will fit in the arrays as dimensioned. The first two of these parameters,  $NDIM$  and  $NDIM1$ , correspond to the dimensions of the field arrays,  $X$ , etc. The last two,  $NDIM2$  and  $NDIM3$ , correspond to the dimensions of the slit arrays,  $XL$ , etc. The last parameter,  $NDIM3$ , also corresponds to the dimension of the segment arrays,  $LB1$ , etc. Thus  $NDIM$  is the maximum value of  $\xi$  that can be used, while  $NDIM1$  is the maximum value  $\eta$  allowed. Also,  $NDIM2$  is the maximum number of points that can be used on a slit or slab side, and  $NDIM3$  is

the maximum number of slits and slab sides that can be used. The input thus must satisfy the following:

$$IMAX \leq NDIM$$

$$JMAX \leq NDIM1$$

$$|LB2(L) - LB1(L)| + 1 \leq NDIM2 \quad L = 1, 2, \dots, NBDY$$

$$NBDY \leq NDIM3$$

After the initial input parameters are read, the code does some setup of various intermediate parameters and checks for compatibility with the array dimensions. The value of IDISK is then checked to determine if the solution is to be started from the beginning or if a partially converged solution is to be continued.

Boundary input and construction. If the start is from the beginning, the point type array LSLIT is initialized to -20000 on the outer rectangle formed by  $I = 1$  &  $IMAX$  and  $J = 1$  &  $JMAX$ , and to 0 inside this rectangle.

Next the points on the slits and/or slab sides (if any) are read from either card images or file 10. Points on slits are placed in the slit arrays, XL, etc., while points on slab sides are placed directly in the field arrays X and Y. The point type array LSLIT is set to  $-(10000 + L)$  at points on slab sides, where L identifies the particular segment in the order as input, unless the side is a part of the outer boundary in which case LSLIT is left at -20000. At the same time, 10 is added to the segment type array LTYPE for slab sides on the outer boundary, resulting in replacing the input values of -11 and -12 with -1 and -2, respectively, in conformance with the usage for slits.

The slit arrays, XL, etc. (if any), are then printed and subroutine BNDRY is called for the outer boundary. If the outer boundary is not

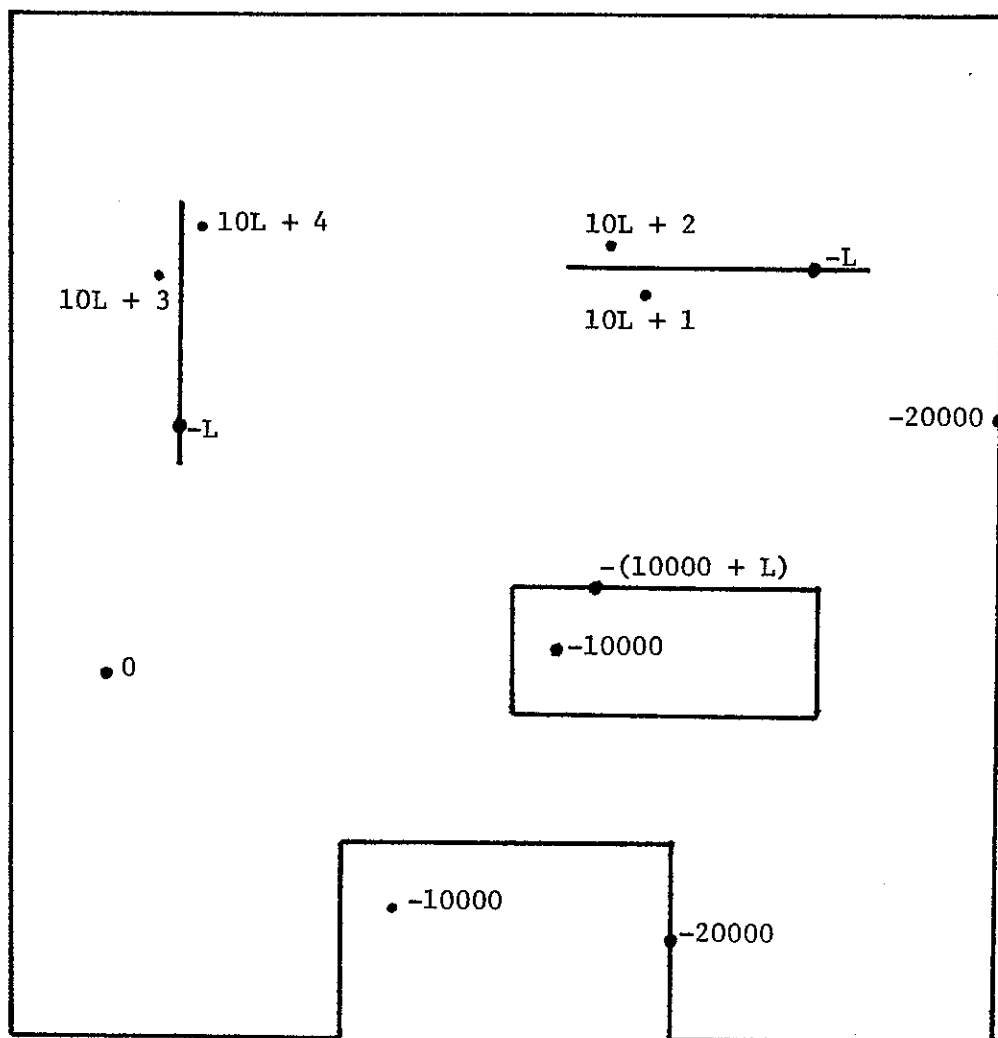


input in segments as slab sides, this boundary is either input as a succession of points proceeding from a specified point completely around the outer rectangle formed by  $I = 1, I_{MAX}$ , and  $J = 1, J_{MAX}$ , or a circular outer boundary is generated internally and placed on this rectangle. Both of these procedures are performed by this subroutine by calling the subroutine INFBDY, which either reads a point from a card image or calculates a point on the circle.

Point types. Next the point type array LSLIT is set to the following values on and adjacent to slits (if any). Here L identifies the particular slit in the order as input:

-L	:	on slit	
10L + 1	:	below horizontal slit	not adjacent to slit ends
10L + 2	:	above horizontal slit	
10L + 3	:	left of vertical slit	
10L + 4	:	right of vertical slit	

The point type array LSLIT is then set to -10000 for points outside the computational region, i.e., inside slabs, by sweeping along each  $\xi$ - and  $\eta$ -line and noting when the computational region is entered or left across a slab side. The complete point type array then contains the values indicated in the following diagram:



Control functions and initial guess. With all of the boundary points in place and the point type array filled, the code then calls subroutine `CONTRL` to evaluate the control functions on the entire boundary (including internal boundaries). The subroutine `GUESSA` is called next to calculate the control functions and the initial guess for the cartesian coordinates in the field by interpolation from the values on the boundaries. This interpolation is done at each point in the field by locating the pair of

boundary segments facing the point (one or both members may be internal boundaries) and interpolating between these segments. For the coordinate values, the distances separating the pair of segments facing the point in the horizontal and vertical directions are examined and the interpolation is done between the pair with the smaller separation.

Iterative solution. If the solution is to be restarted from a partially converged result, then all of the above computations are skipped and the partially converged solution is read from file 10 instead. In either case the initial array values are printed at this point according to the input print controls.

Subroutine TRANS is now called to perform the iterative solution. This subroutine first reads the parameters associated with the attraction of curvilinear coordinate lines to other curvilinear coordinate lines and/or points. The species of line being controlled, i.e.,  $\xi$  or  $\eta$ , is read into ATYP, and whether the control is to be attraction or repulsion is determined by the input parameter ITYP. The number of coordinate lines and points designated as sources of attraction are read into NLN and NPT, respectively. Also, a common decay factor and a common amplitude multiplication factor to be used for all attraction lines and points for this species can be read into DEC and AMPFAC, respectively.

For each species of control, subroutine RHS is called to read the attraction line index, or point indices, and the amplitude and decay factor for each. This subroutine also sums the effects for all such attraction lines and points and adds this cumulative effect to the control function at each point in the field in accordance with Eq. (5) of Part A.

Subroutine TRANS then reads the parameters associated with attraction of curvilinear coordinate lines to specified lines and/or points in space and adds the cumulative effect of all such attraction lines and/or points to the control functions at each point in the field. This is done in a similar manner as described above. Subroutine RHSXY reads the cartesian coordinates of the points on the specified attraction line and those of the isolated attraction points and calculates the normal to the attraction line. These quantities are placed in the arrays XPT, YPT, VEC1, and VEC2. The addition to the control functions in this case must be changed as the iterative solution of x and y proceeds since the control functions depend on x and y for this type of attraction.

After completing the calculation of the control functions, subroutine TRANS reads the parameters that provide for a gradual implementation of these equations during the iteration, and performs some setup for the iterative solution.

The field is then swept iteratively until convergence is achieved or the maximum number of iterations allowed is reached. In each iteration, new values for x and y at points having the point type LSLIT non-negative are calculated.

First, the coordinate derivatives are calculated, and the Jacobian and other such quantities and coefficients are evaluated. Then the locally optimum acceleration parameters are calculated if such is elected. The change in these acceleration parameters between iterations is monitored and the values are frozen when the magnitude of the change falls below a specified tolerance at all points. (This change between iterations, and the analogous changes in x and y, are calculated by calling subroutine ERROR.) The acceleration parameter is placed in the field

array WACC. The addition to the control functions from attraction to specified lines and/or points in the physical region is calculated next, and then the new values of x and y for the point are calculated.

This procedure is followed for all points in the field, i.e., points having the point type LSLIT non-negative. For points adjacent to slits it is necessary to obtain the values on the slit from the slit arrays, XL, etc., and the calculations are done in that case by calling subroutine SLIT.

After each sweep of the field the maximum changes in x and y from the previous sweep are compared with the input tolerances. If the maximum number of iterations allowed by the input is reached before convergence, then the partially converged solution is written on file 10 for potential restart. If convergence is obtained, the solution is written on file 11.

#### LINES (Boundary Segments)

The small front-end code LINES generates a distribution of a specified number of points on a curve between two specified points. The curve may be specified to be a straight line, a circular or elliptic arc, a quadratic with zero slope at either end point, or a cubic with the slope specified at both ends. In any case the point distribution on the curve may be uniform or exponentially concentrated toward either end. The input consists of the number of curves to be generated and, for each curve, the number of points on the curve, the type of curve, the end points, and the particular quantities to be specified in connection with each curve. Detailed instructions for input are given below.

The cartesian coordinates of the points generated on each curve are output in succession on file 10 by a separate unformatted write statement for each point (WRITE(10) X,Y). Since more than one curve can be generated in one run, this code can be used to build an entire boundary composed of segments of different types. The generation of the curves and the exponential concentration of points thereon are explained in the following section.

### Generation of Curves

Straight line. Here we have simply

$$y = a + bx$$

so that with the end points  $(x_1, y_1)$  and  $(x_2, y_2)$  specified we have

$$\begin{pmatrix} 1 & x_1 \\ 1 & x_2 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

so that

$$a = \frac{y_1 x_2 - y_2 x_1}{x_2 - x_1}$$

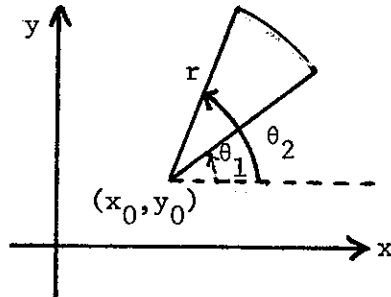
$$b = \frac{y_2 - y_1}{x_2 - x_1}$$

Circular arc. For a circular arc of radius  $r$  centered at  $(x_0, y_0)$  with  $\theta$  measured counter-clockwise from the positive x-axis, we have

$$x = x_0 + r \cos \theta$$

$$y = y_0 + r \sin \theta$$

The end points are defined by inputting the radius  $r$  and center of the arc  $(x_0, y_0)$ , together with the angles  $\theta_1$  and  $\theta_2$  of the end points.



Elliptic arc. In this case we have, for an ellipse with semi-major axis,  $a$ , and semi-minor axis,  $b$ , centered at  $x_0, y_0$ , the equation

$$\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} = 1$$

which can be written in terms of the angle  $\theta$ , measured counter-clockwise from the positive  $x$ -axis, and the angular-dependent radius  $r(\theta)$  as

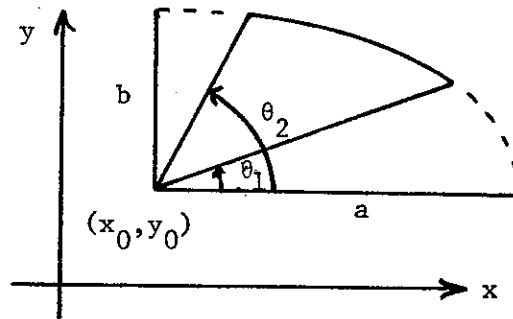
$$x = x_0 + r(\theta) \cos \theta$$

$$y = y_0 + r(\theta) \sin \theta$$

Then

$$r(\theta) = \left| \frac{\cos^2 \theta}{a^2} + \frac{\sin^2 \theta}{b^2} \right|^{-1/2}$$

The end points are specified by inputting the axes a and b, the center  $(x_0, y_0)$ , and the angles of the end points.



Quadratic with zero slope at end point. Here we have

$$y = a + bx + cx^2$$

$$y' = b + 2cx$$

Then with the end points  $(x_1, y_1)$  and  $(x_2, y_2)$  specified together with the specification of zero slope at end point  $i$  ( $i = 1$  or  $2$ ) we have

$$\begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 0 & 1 & 2x_i \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ 0 \end{pmatrix}$$

which is solved for the coefficients a, b, c.

Cubic. The cubic equation is

$$y = a + bx + cx^2 + dx^3$$

$$y' = b + 2cx + 3dx^2$$

or



$$\begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 0 & 1 & 2x_1 & 3x_1^2 \\ 0 & 1 & 2x_2 & 3x_2^2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_1' \\ y_2' \end{pmatrix}$$

which is solved for the coefficients a, b, c, d.

### Exponential Concentration of Points

The exponential distribution of points on the curve of any type is done by taking

$$x_n = x_1 + (x_2 - x_1) \left[ 1 - \frac{1 - e^{-\alpha(N-n)}}{1 - e^{-\alpha(N-1)}} \right]$$

for concentration near the first end point and

$$x_n = x_1 + (x_2 - x_1) \left[ \frac{1 - e^{-\alpha(n-1)}}{1 - e^{-\alpha(N-1)}} \right]$$

for concentration near the second end point. Here the strength of the concentration is controlled by the specified decay factor  $\alpha$ , and N is the number of points on the curve.

## CSPLØT (Plot)

The plot code CSPLØT plots the coordinate system generated by the code WESCØR, having read the coordinate system from file 11 as output by WESCØR. The input consists of the number of coordinate lines to be plotted, a designation for skipping lines, the extent of the field to be plotted, and a factor for using different seating in the horizontal and vertical directions. This input is detailed in the following pages.

The plot is formed by simply connecting the points on a line of constant curvilinear coordinates in the physical region, i.e., by constructing straight lines between each successive pair of points,  $X(I,J)$  and  $Y(I,J)$ , as one index is held fixed.

# WESCØR INPUT INSTRUCTIONS

```

.00=C*****
110=C
120=C***** W E S C Ø R *****
130=C
140=C*****
150=C
160=C 2-D BOUNDARY-FITTED COORDINATE SYSTEM CODE
170=C
180=C MISSISSIPPI STATE UNIVERSITY , 1982
190=C
200=C U.S. ARMY ENGINEER WATERWAYS EXPERIMENT STATION
210=C VICKSBURG, MISSISSIPPI
220=C
230=C*****
240=C
250=C ***** SLIT-SLAB CONFIGURATION *****
260=C *
270=C **** ATTRACTION TO COORDINATE LINES/POINTS AND TO SPACE LINES/POINTS.
280=C **** CONTROL FUNCTIONS ALSO INTERPOLATED FROM BOUNDARY POINT DISTRIBUTION.
290=C *
300=C *****
310=C ***** INPUT INSTRUCTIONS :
320=C *
330=C *** CARDS(2) : LABEL - FORMAT(10A8)
340=C *
350=C * LABEL - TWO 80 CHARACTER CARDS. (BLANK CARDS IF NO LABEL)
360=C *
370=C *** CARD : IMAX,JMAX,NBDY,ITER,ISLIT,IBNDY,IDISK,IWIR,IWINTL,
380=C * IWIN,NREN - FORMAT(11I5)
390=C *
400=C * IMAX - NUMBER OF XI POINTS.
410=C *
420=C * JMAX - NUMBER OF ETA POINTS.
430=C *
440=C * NBDY - TOTAL NUMBER OF SLAB SIDES AND SLITS IN THE FIELD.
450=C *
460=C * ITER - MAXIMUM NUMBER OF ITERATIONS ALLOWED.
470=C *
480=C * ISLIT - =1 SLAB SIDES OR SLITS READ FROM CARDS.
490=C * X,Y - FORMAT(2F10.0) , ONE POINT PER CARD.
500=C * =2 SLAB SIDES OR SLITS READ FROM FILE 10.
510=C * X,Y - UNFORMATTED , ONE POINT PER IMAGE.
520=C *
530=C * (NOTE: HORIZONTAL SLITS ARE READ CLOCKWISE FROM RIGHT END.)
540=C * ( VERTICAL SLITS ARE COUNTER-CLOCKWISE FROM TOP. )
550=C * ( SLAB SIDES MAY BE READ IN EITHER DIRECTION. )
560=C *
570=C * IBNDY - =0 OUTER BOUNDARY CALCULATED INTERNALLY AS CIRCLE.
580=C * =1 OUTER BOUNDARY READ FROM CARDS.
590=C * X,Y - FORMAT(2F10.0) , ONE POINT PER CARD.
600=C * =2 OUTER BOUNDARY READ FROM FILE 10.
610=C * X,Y - UNFORMATTED , ONE POINT PER IMAGE.
620=C * =-1 OUTER BOUNDARY READ IN SEGMENTS AS SLAB SIDES.
630=C *
640=C * (NOTE: FOR IBNDY = 1 OR 2 , OUTER BOUNDARY IS READ CLOCKWISE)
650=C * ( FROM POINT (INFXI,INFETA). )
660=C * ( "OUTER BOUNDARY" MEANS ENTIRE BOUNDARY OF TRANSFORMED )
670=C * ( REGION IF NREN=0. IF NREN IS NOT ZERO, THEN OUTER )
680=C * ( BOUNDARY IS THE TOP OF THE TRANSFORMED REGION AND )
690=C * ( INNER BOUNDARY IS THE BOTTOM. )
700=C *
710=C * IDISK - =0 DON'T READ OR WRITE SYSTEM FROM OR ON FILE.
720=C * =1 WRITE SYSTEM ON FILES 10 & 11. DON'T READ SYSTEM FROM FILE.
730=C * =2 WRITE SYSTEM ON FILES 10 & 11 , READ SYSTEM FROM FILE 10 FOR RESTART.
740=C * =3 READ SYSTEM FROM FILE 10 FOR RESTART, DON'T WRITE SYSTEM ON FILE 11.
750=C *
760=C * (NOTE: FILE 10 IS RESTART FILE FOR CONTINUATION OF ITERATION.)
770=C * ( FILE 11 IS STORAGE FILE FOR FINAL SYSTEM. )
780=C *
790=C * IWIR - =0 DON'T PRINT EACH ITERATION ERROR.
800=C * =1 PRINT EACH ITERATION ERROR.
810=C *
820=C * IWINTL - =0 DON'T PRINT INITIAL GUESS.
830=C * =1 PRINT INITIAL GUESS.
840=C *
850=C * IWIN - NON-ZERO SUPPRESSES PRINT OF FINAL VALUES.

```

```

860=C *
870=C *
880=C *   NREN - NON-ZERO USES RE-ENTRANT BOUNDARY ON LEFT & RIGHT SIDES
890=C *   OF TRANSFORMED REGION, WITH OUTER BOUNDARY ON TOP
900=C *   AND INNER BOUNDARY ON BOTTOM.
910=C *
920=C *   INNER BOUNDARY IS READ AS FOLLOWS BEFORE READING OUTER BOUNDARY:
930=C *   =1 INNER BOUNDARY READ FROM CARDS.
940=C *     X,Y - FORMAT(2F10.0), ONE POINT PER CARD.
950=C *   =2 INNER BOUNDARY READ FROM FILE 10.
960=C *     X,Y - UNFORMATTED, ONE IMAGE PER CARD.
970=C *
980=C *   (NOTE: SLITS AND/OR SLABS MAY ALSO BE PRESENT.)
990=C *
1000=C *   *** CARDS(NBODY) LB1, LB2, LB3, LTYPE - FORMAT(4I5)
1010=C *
1020=C *   LB1, LB2 - FIRST AND LAST INDICES OF SLAB SIDE OR SLIT ENDS.
1030=C *             (LB2 MAY BE LESS THAN LB1 FOR SLAB SIDE. INPUT IS FROM LB1 TO LB2.)
1040=C *
1050=C *   LB3 - INDEX OF LINE ON WHICH SLAB SIDE OR SLIT IS LOCATED.
1060=C *
1070=C *   LTYPE - SLAB SIDE OR SLIT TYPE (1 FOR HORIZONTAL, 2 FOR VERTICAL.)
1080=C *
1090=C *   (NEGATIVE INDICATES SLAB SIDE, RATHER THAN SLIT.)
1100=C *   (SUBTRACT 10 FOR OUTER BOUNDARY SEGMENT.)
1110=C *   (I.E., -11 IS HORIZONTAL OUTER BOUNDARY SEGMENT,)
1120=C *   ( -12 IS VERTICAL OUTER BOUNDARY SEGMENT. )
1130=C *
1140=C *   *** CARD R(1), R(2), R(3), YINFIN, AINFIN, X0INF, Y0INF, INF1, INFETA
1150=C *   - FORMAT(7F10.0, 2I5)
1160=C *
1170=C *   R(1) - SOURCE ACCELERATION PARAMETER.
1180=C *   (ZERO VALUE CAUSES VARIABLE ACCELERATION PARAMETER)
1190=C *   (FIELD TO BE CALCULATED INTERNALLY. )
1200=C *
1210=C *   R(2) - ALLOWABLE X ITERATION ERROR.
1220=C *
1230=C *   R(3) - ALLOWABLE Y ITERATION ERROR.
1240=C *
1250=C *   YINFIN - RADIUS OF CIRCULAR OUTER BOUNDARY.
1260=C *
1270=C *   AINFIN - ANGLE OF FIRST POINT ON CIRCULAR OUTER BOUNDARY (DEGREES).
1280=C *   (COUNTER-CLOCK FROM POSITIVE X-AXIS.)
1290=C *
1300=C *   X0INF, Y0INF - CENTER OF CIRCULAR OUTER BOUNDARY.
1310=C *
1320=C *   NINF - NUMBER OF UNIQUE POINTS ON CIRCULAR OUTER BOUNDARY.
1330=C *
1340=C *   INF1, INFETA - INDICES OF FIRST POINT ON CIRCULAR OUTER BOUNDARY.
1350=C *
1360=C *   (NOTE : LAST 7 OF THESE PARAMETERS ARE IRRELEVANT IF OUTER BOUNDARY IS READ.)
1370=C *
1380=C *
1390=C *   ** IF BODIES AND/OR OUTER BOUNDARY ARE READ FROM CARDS, SUCH CARDS
1400=C *   FOLLOW NEXT.
1410=C *
1420=C *   ** SLITS AND/OR SLAB SIDES ARE READ FIRST, THEN OUTER BOUNDARY IS READ.
1430=C *   (THESE RULES APPLY FOR READING FROM FILE 10 AS WELL AS FROM CARDS.)
1440=C *
1450=C *
1460=C *
1470=C *   ** IF NO COORDINATE ATTRACTION IS TO BE USED, FOLLOW THESE CARDS
1480=C *   WITH FIVE BLANK CARDS. IF ATTRACTION IS TO BE USED, USE THE FOLLOWING
1490=C *   INPUT RATHER THAN THE BLANK CARDS:
1500=C *
1510=C *   ** INPUT FOR COORDINATE SYSTEM CONTROL : USE FOUR SETS, ONE FOR
1520=C *   ** XI-LINE ATTRACTION TO COORDINATE LINES/POINTS, ONE FOR ETA-LINE ATTRACTION
1530=C *   ** TO COORDINATE LINES/POINTS, ONE FOR XI-LINE ATTRACTION TO SPACE LINES/POINTS,
1540=C *   ** AND ONE FOR ETA-LINE ATTRACTION TO SPACE LINES/POINTS.
1550=C *   ** ANY SET NOT WANTED IS REPLACED BY ONE BLANK CARD.
1560=C *
1570=C *   ****
1580=C *
1590=C *   ** THE FOLLOWING, MARKED WITH #, IS FOR ATTRACTION TO COORDINATE LINES/POINTS:
1600=C *
1610=C *   # *** CARD : ATYP, ITYP, N, M, NPT, DEC, AMPFAC - FORMAT(A8, I2, 2I5, 2F10.0)
1620=C *
1630=C *   #   ATYP - TYPE OF ATTRACTION. (X1 FOR XI-LINE ATTRACTION,
1640=C *   #   ETA FOR ETA-LINE ATTRACTION.) LEFT JUSTIFIED.
1650=C *

```

```

1660=C **      ITYP  - ZERO GIVES ATTRACTION ON BOTH SIDES.
1670=C **      NON-ZERO GIVES ATTRACTION ON UPPER SIDE AND
1680=C **      REPUSSION ON LOWER SIDE.
1690=C **
1700=C **      NLN   - NUMBER OF ATTRACTION LINES.
1710=C **
1720=C **      NPT   - NUMBER OF ATTRACTION POINTS.
1730=C **
1740=C **      DEC   - NON-ZERO DEC USES DEC FOR DECAY FACTOR.
1750=C **
1760=C **      AMPFAC - NON-ZERO AMPFAC MULTIPLIES ALL AMPLITUDES BY AMPFAC.
1770=C **
#2 1780=C **** CARDS(NLN) : JLN,ALN,DLN - FORMAT(5X,I5,2F10.0)
1790=C **      (OMIT IF NLN IS ZERO)
1800=C **
1810=C **      JLN - ATTRACTION LINE INDEX.
1820=C **
1830=C **      ALN - AMPLITUDE (NEGATIVE REPELS) FOR LINE ATTRACTION.
1840=C **
1850=C **      DLN - DECAY FACTOR FOR LINE ATTRACTION.
1860=C **
1870=C **** CARDS(NPT) : IPT,JPT,APT,DPT - FORMAT(2I5,2F10.0)
1880=C **      (OMIT IF NPT IS ZERO)
1890=C **
1900=C **      IPT,JPT - ATTRACTION POINT INDICES.
1910=C **
1920=C **      APT - AMPLITUDE (NEGATIVE REPELS) FOR POINT ATTRACTION.
1930=C **
1940=C **      DPT - DECAY FACTOR FOR POINT ATTRACTION.
1950=C **
1960=C ****
1970=C **
1980=C *** THE FOLLOWING, MARKED WITH $, IS FOR ATTRACTION TO SPACE LINES/POINTS :
1990=C **
2000=C **** THE FOLLOWING CARDS ARE FOR ATTRACTION TO LINES AND/OR POINTS
2010=C **** DEFINED BY X,Y COORDINATES. IF NLN IS NOT ZERO, THEN NLN
2020=C **** OF THE CARDS GIVING NP MUST APPEAR. EACH OF THESE CARDS IS
2030=C **** FOLLOWED BY NP OF THE CARDS GIVING XPT, ETC. IF NPT IS NOT
2040=C **** ZERO, THEN NPT OF THE CARDS GIVING XPT, ETC. MUST FOLLOW
2050=C **** THE LAST GROUP OF THESE CARDS.
2060=C **** ANY SET NOT WANTED IS REPLACED BY ONE BLANK CARD.
2070=C **
#3 2080=C **** CARD : ATYP,ITYP,NLN,NPT,DEC,AMPFAC - FORMAT(A8,I2,2I5,2F10.0)
2090=C **
2100=C **      ATYP  - TYPE OF ATTRACTION. (XI FOR XI-LINE ATTRACTION,
2110=C **      ETA FOR ETA-LINE ATTRACTION.) LEFT JUSTIFIED.
2120=C **
2130=C **      ITYP  - ZERO GIVES ATTRACTION ON BOTH SIDES.
2140=C **      NON-ZERO GIVES ATTRACTION ON UPPER SIDE AND
2150=C **      REPUSSION ON LOWER SIDE.
2160=C **
2170=C **      NLN   - NUMBER OF ATTRACTION LINES.
2180=C **
2190=C **      NPT   - NUMBER OF ATTRACTION POINTS.
2200=C **      (NOT INCLUDING POINTS ON ATTRACTION LINES)
2210=C **
2220=C **      DEC   - NON-ZERO DEC USES DEC FOR DECAY FACTOR.
2230=C **
2240=C **      AMPFAC - NON-ZERO AMPFAC MULTIPLIES ALL AMPLITUDES BY AMPFAC.
2250=C **
#4 2260=C **** CARD : NP - FORMAT(I5)
2270=C **
2280=C **      NP - NUMBER OF POINTS ON THIS ATTRACTION LINE.
2290=C **
2300=C **** CARDS : XPT,YPT,APT,DPT,VEC1,VEC2 - FORMAT(6F10.0)
2310=C **
2320=C **      XPT,YPT - COORDINATES OF ATTRACTION POINT OR
2330=C **      POINT ON ATTRACTION LINE.
2340=C **
2350=C **      APT - ATTRACTION AMPLITUDE (NEGATIVE REPELS).
2360=C **
2370=C **      DPT - DECAY FACTOR.
2380=C **
2390=C **      VEC1,VEC2 - X,Y COMPONENTS OF UNIT VECTOR NORMAL TO
2400=C **      ATTRACTION DIRECTION FOR POINT ATTRACTION.
2410=C **      (CALCULATED INTERNALLY FOR LINE ATTRACTION.)
2420=C **
2430=C **
2440=C ** .....
2450=C **

```

```

2460=C *** THE LAST COORDINATE SYSTEM CONTROL CARD IS THE FOLLOWING CARD :
2470=C *
2480=C *** CARD : IFAC,IRIT,EFAC - FORMAT(2I5,F10.0)
2490=C *
2500=C *      (CAN BE USED TO AID CONVERGENCE BY CONVERGING FIELD )
2510=C *      (WITH LESS ATTRACTION FIRST AND USING THIS RESULT )
2520=C *      (AS THE INITIAL GUESS FOR STRONGER ATTRACTION. )
2530=C *      (BLANK CARD MUST BE INPUT IF THIS FEATURE IS NOT USED.)
2540=C *      (STANDARD IS TO NOT USE THIS FEATURE , BUT ITS USE MAY)
2550=C *      (BE NECESSARY WITH STRONG ATTRACTION. )
2560=C *
2570=C *      IFAC - NUMBER OF STEPS IN ADDITION OF INHOMOGENEOUS TERM,
2580=C *      DOUBLES INHOMOGENEOUS TERM AT EACH STEP.
2590=C *
2600=C *      (ZERO CONVERGES WITH FULL ATTRACTION. )
2610=C *      (1.0 CONVERGES WITH NO ATTRACTION FIRST, THEN )
2620=C *      (WITH FULL ATTRACTION. 2.0 CONVERGES WITH NO )
2630=C *      (ATTRACTION FIRST, THEN WITH HALF, THEN WITH FULL.)
2640=C *      (INCREASE NUMBER OF STEPS IF DIVERGENCE OCCURS. )
2650=C *
2660=C *      IRIT - NON-ZERO VALUE CAUSES INHOMOGENEOUS TERM TO BE PRINTED.
2670=C *
2680=C *      EFAC - MULTIPLE OF CONVERGENCE CRITERION TO BE USED FOR
2690=C *      INTERMEDIATE CONVERGENCE BETWEEN ADDITIONS OF
2700=C *      INHOMOGENEOUS TERM. (TYPICALLY 10.0 )
2710=C *
2720=C *****
2730=C *
2740=C * MASS STORAGE FILES :
2750=C *
2760=C * RESTART FILE - FILE 10 :
2770=C *
2780=C *      (10) RXI,RETA
2790=C *      (10) X,Y,LSLIT,LABEL,IMAX,JMAX
2800=C *      (10) NBDY,NUMB,LB1,LB2,LB3,LTYPE,LPT,XL,XU,YL,YU,
2810=C *      NDIM,NDIM1,NDIM2,NDIM3,WACC
2820=C *
2830=C * COORDINATE SYSTEM STORAGE FILE - FILE 11 :
2840=C *
2850=C *      (11) LABEL,IMAX,JMAX
2860=C *      (11) ((LSLIT(I,J),I=1,IMAX),J=1,JMAX)
2870=C *      (11) ((X(I,J),I=1,IMAX),J=1,JMAX)
2880=C *      (11) ((Y(I,J),I=1,IMAX),J=1,JMAX)
2890=C *      (11) NBDY,NUMB,LB1,LB2,LB3,LTYPE,LPT,XL,XU,YL,YU,
2900=C *      NDIM,NDIM1,NDIM2,NDIM3
2910=C *
2920=C *****

```

# LINES INPUT INSTRUCTIONS

```

100=C*****
110=C
120=C***** L I N E S *****
130=C
140=C*****
150=C
160=C BOUNDARY SEGMENT CODE FOR INPUT TO WESCOR
170=C
180=C MISSISSIPPI STATE UNIVERSITY , 1982
190=C
200=C U.S.ARMY ENGINEER WATERWAYS EXPERIMENT STATION
210=C VICKSBURG, MISSISSIPPI
220=C
230=C*****
240=C
250=C*** POINTS ON BOUNDARY SEGMENTS ***
260=C
270=C*****
280=C
290=C*** INPUT :
300=C
310=C**CARD : N LINES - FORMAT(I5)
320=C
330=C N LINES - TOTAL NUMBER OF LINES.
340=C
350=C**CARDS(N LINES) : N, ITYP, D1, D2, D3, D4, D5, D6, DE - FORMAT(2I5, 7F10.0)
360=C
370=C N - NUMBER OF POINTS ON LINE.
380=C
390=C ITYP - TYPE OF LINE
400=C 0 : STRAIGHT.
410=C 1 : CIRCULAR ARC.
420=C 2 : ELLIPTIC ARC.
430=C 3 : CURIC.
440=C 4 : QUADRATIC WITH ZERO SLOPE AT FIRST POINT.
450=C 5 : QUADRATIC WITH ZERO SLOPE AT SECOND POINT.
460=C
470=C D1-D6 AS FOLLOWS - (ITEMS NOT CITED ARE IRRELEVANT)
480=C
490=C ITYP=0 : D1 - X OF FIRST POINT.
500=C D2 - Y OF FIRST POINT.
510=C D3 - X OF SECOND POINT.
520=C D4 - Y OF SECOND POINT.
530=C
540=C ITYP=1 : D1 - ANGLE OF FIRST POINT (DEGREES, COUNTER-CLOCK FROM POSITIVE X-AXIS)
550=C D2 - ANGLE OF SECOND POINT (DEGREES, COUNTER-CLOCK FROM POSITIVE X-AXIS)
560=C D3 - X OF CIRCLE CENTER.
570=C D4 - Y OF CIRCLE CENTER.
580=C D5 - CIRCLE RADIUS.
590=C
600=C ITYP=2 : D1 - ANGLE OF FIRST POINT, (DEGREES, COUNTER-CLOCK FROM POSITIVE X-AXIS)
610=C D2 - ANGLE OF SECOND POINT, (DEGREES, COUNTER-CLOCK FROM POSITIVE X-AXIS)
620=C D3 - X OF ELLIPSE CENTER.
630=C D4 - Y OF ELLIPSE CENTER.
640=C D5 - X-AXIS LENGTH OF ELLIPSE.
650=C D6 - Y-AXIS LENGTH OF ELLIPSE.
660=C
670=C ITYP=3 : D1-D4 SAME AS ITYP=0
680=C D5 - SLOPE AT FIRST POINT, (DEGREES, COUNTER-CLOCK FROM POSITIVE X-AXIS)
690=C D6 - SLOPE AT SECOND POINT, (DEGREES, COUNTER-CLOCK FROM POSITIVE X-AXIS)
700=C
710=C ITYP=4 : D1-D4 SAME AS ITYP=0
720=C
730=C ITYP=5 : D1-D4 SAME AS ITYP=0
740=C
750=C DE - EXPONENTIAL CONCENTRATION FACTOR.
760=C 0.0 FOR EQUAL SPACING ON LINE.
770=C NEGATIVE FOR CONCENTRATION NEAR FIRST POINT.
780=C POSITIVE FOR CONCENTRATION NEAR SECOND POINT.
790=C
800=C*****
810=C
820=C MASS STORAGE FILE :
830=C
840=C OUTPUT - FILE 10 : WRITE(10) X(I),Y(I)
850=C X & Y POINTS OF EACH LINE, INCLUDING ENDS.

```

# CSPLØT INPUT INSTRUCTIONS

```

100=C*****
110=C*
120=C***** C S P L O T *****
130=C*
140=C*****
150=C*
160=C  COORDINATE SYSTEM PLOT CODE - MISSISSIPPI STATE UNIVERSITY , 1982
170=C
180=C          U.S. ARMY ENGINEER WATERWAYS EXPERIMENT STATION
190=C          VICKSBURG, MISSISSIPPI
200=C*
210=C*****
220=C
230=C *****
240=C *
250=C ***** INPUT INSTRUCTIONS :
260=C *
270=C *** CARD : NUMBR , NUMBR1 , ISKIP1 , ISKIP2 - FORMAT(4I5)
280=C *
290=C *      NUMBR - NUMBER OF ETA=CONSTANT LINES DESIRED FOR PLOT.
300=C *      (DEFAULT IS ALL LINES)
310=C *
320=C *      NUMBR1 - NUMBER OF XI=CONSTANT LINES DESIRED FOR PLOT.
330=C *      (DEFAULT IS ALL LINES)
340=C *
350=C *      ISKIP1 - SKIP PARAMETER FOR XI=CONSTANT COORDINATE LINES.
360=C *      (1 PLOTS EVERY LINE, 2 PLOTS EVERY SECOND LINE, ETC.)
370=C *      (DEFAULT IS EVERY LINE)
380=C *
390=C *      ISKIP2 - SKIP PARAMETER FOR ETA=CONSTANT COORDINATE LINES.
400=C *      (SEE ISKIP1)
410=C *
420=C *** CARD : IB1 , IB2 , JB1 , JB2 - FORMAT(4I5)
430=C *
440=C *      I,J INDICES OF PLOT FIELD BOUNDARY.
450=C *      (I IS XI, J IS ETA. DEFAULT IS ENTIRE FIELD)
460=C *
470=C *** CARD : XYRAT - FORMAT(F10.0)
480=C *
490=C *      XYRAT - RATIO OF PLOTTED X TO Y LENGTHS. (1.0)
500=C *
510=C *****
520=C *
530=C ***** COORDINATE SYSTEM IS READ UNFORMATTED FROM UNIT 10 AS
540=C ***** WRITTEN BY THE CODE 'WESCOR'
550=C *
560=C *****

```



## SAMPLE RUNSTREAMS

### LINES Sample Runstream #1

```

120=JOB.
130=ACQUIRE,DN=BINARY,PDN=THOMPSONLINESB,ID=████████,DF=TR,UQ.
140=LDR,DN=BINARY.
150=DISPOSE,DN=FT10,SDN=FILE,ID=████████,DC=ST,DF=TR,TEXT=↑
160= 'CATALOG,FILE,THOMPSONLINESD,ID=████████,RP=999.'.
170=DELETE,DN=BINARY.
180=EXIT.
190=DELETE,DN=BINARY.
200=$EOR
210= 5
220= 33 0 0.0 0.0 24.39 0.0
230= 9 0 0.0 -0.3 6.1 -0.3
240= 25 0 6.1 -0.3 24.39 -0.91
250= 25 0 0.0 -0.3 0.0 0.0
260= 25 0 24.39 -0.91 24.39 0.0
270=$EOR
280=$EOF

```

### WESCOR Sample Runstream #1

```

120=JOB,T=60.
130=ACQUIRE,DN=FT10,PDN=THOMPSONLINESD,ID=████████,DF=TR,UQ.
140=ACQUIRE,DN=BINARY,PDN=THOMPSONCORDB,ID=████████,DF=TR,UQ.
150=LDR,DN=BINARY.
160=DISPOSE,DN=FT11,SDN=FILE,ID=████████,DC=ST,DF=TR,TEXT=↑
170= 'CATALOG,FILE,THOMPSONCORDD,ID=████████,RP=999.'.
180=DELETE,DN=FT10.
190=DELETE,DN=BINARY.
200=EXIT.
210=DELETE,DN=FT10.
220=DELETE,DN=BINARY.
230=$EOR
240=JOHNSON FLUME
250=33 X 25 COORDINATE SYSTEM
260= 33 25 5 100 2 -1 1 1 1 0 0
270= 1 33 25 -11
280= 1 9 1 -11
290= 9 33 1 -11
300= 1 25 1 -12
310= 1 25 33 -12
320=1.8 0.00001 0.00001
330=
340=
350=
360=
370=
380=$EOR
390=$EOF

```

### CSPLØT Sample Runstream

```

120=JOB.
130=ACQUIRE,DN=FT10,PDN=THOMPSONCORDD,ID=████████,DF=TR,UQ.
140=ACQUIRE,DN=BINARY,PDN=THOMPSONCSPLØT,ID=████████,DF=TR,UQ.
150=LDR,LIB=METALIB,DN=BINARY.
160=DISPOSE,DN=FT01,SDN=FILE,ID=████████,DC=ST,DF=SB,TEXT=↑
170= 'CATALOG,FILE,THOMPSONPLOT,ID=████████,RP=999.'.
180=DELETE,DN=FT10.
190=DELETE,DN=BINARY.
200=EXIT.
210=DELETE,DN=FT10.
220=DELETE,DN=BINARY.
230=$EOR
240= 0 0 1 1
250= 0 0 0 0
260=2.0
270=$EOR
280=$EOF

```

## LINES Sample Runstream #2

```

120=JOB.
130=ACQUIRE,DN=BINARY,PDN=THOMPSONLINESB,ID=████████,DF=TR,UQ.
140=LDR,DN=BINARY.
150=DISPOSE,DN=FT10,SDN=FILE,ID=████████,DC=ST,DF=TR,TEXT=↑
160=      'CATALOG,FILE,THOMPSONLINESB1,ID=████████,RF=999.'
170=DELETE,DN=BINARY.
180=EXIT.
190=DELETE,DN=BINARY.
200=EOF
210=  15
220=  25  0 15.0  0.0  5.91  0.0  0.0  0.0  -0.08
230=  9  0 5.91  0.0  4.1  0.68  0.0  0.0  0.2
240=  3  0 4.1  0.68  3.9  0.68  0.0  0.0  0.0
250=  9  0 3.9  0.68  2.29  0.0  0.0  0.0  -0.2
260=  7  0 2.29  0.0  0.0  0.0  0.0  0.0  0.0
270= 23  0 0.0  0.0  0.0  1.63  0.0  0.0  0.0
280= 21  0 0.0  1.63  5.91  1.63  0.0  0.0  0.0
290= 29  0 5.91  1.63  15.0  1.63  0.0  0.0  0.04
300=  7  0 15.0  1.63  15.0  1.25  0.0  0.0  0.0
310=  5  3 15.0  1.25  15.25  0.96  -57.0  135.0  0.0
320=  5  0 15.25  0.96  15.8  0.43  0.0  0.0  0.0
330=  9  0 15.8  0.43  15.63  0.23  0.0  0.0  0.0
340=  5  0 15.63  0.23  15.25  0.58  0.0  0.0  0.0
350=  5  3 15.25  0.58  15.0  0.5  135.0  51.0  0.0
360=  9  0 15.0  0.5  15.0  0.0  0.0  0.0  0.0
370=EOF
380=EOF

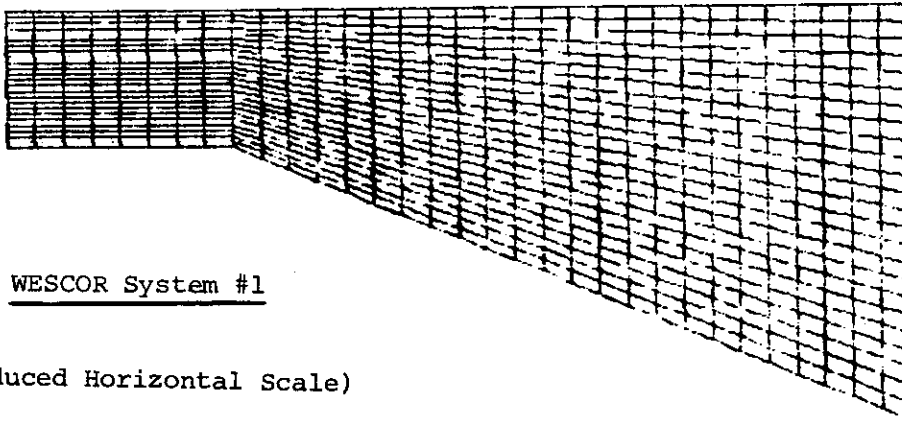
```

## WESCOR Sample Runstream #2

```

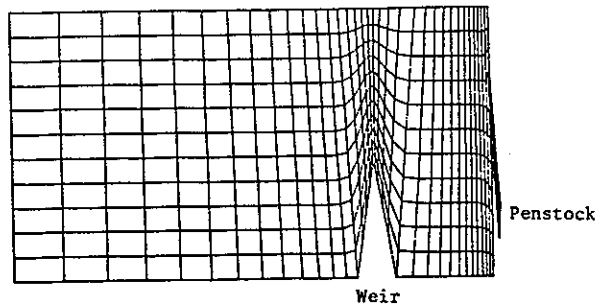
120=JOB,T=60.
130=ACQUIRE,DN=FT10,PDN=THOMPSONLINESD1,ID=████████,DF=TR,UQ.
140=ACQUIRE,DN=BINARY,PDN=THOMPSONCORDB,ID=████████,DF=TR,UQ.
150=LDR,DN=BINARY.
160=DISPOSE,DN=FT11,SDN=FILE,ID=████████,DC=ST,DF=TR,TEXT=↑
170=      'CATALOG,FILE,THOMPSONCORD01,ID=████████,RF=999.'
180=DELETE,DN=FT10.
190=DELETE,DN=BINARY.
200=EXIT.
210=DELETE,DN=FT10.
220=DELETE,DN=BINARY.
230=EOF
240= DORTCH TEST 6 - WITH WEIR
250= 57 X 23 COORDINATE SYSTEM
260=  57  23  15 100  2 -1  1  1  1  0  0
270=  49  25  1 -11
280=  25  17  1 -11
290=  17  15  1 -11
300=  15  7  1 -11
310=  7  1  1 -11
320=  1  23  1 -12
330=  1  21  23 -11
340=  21  49  23 -11
350=  23  17  49 -2
360=  49  53  17 -1
370=  53  57  17 -1
380=  17  9  57 -12
390=  57  53  9 -1
400=  53  49  9 -1
410=  9  1  49 -2
420=1.8      0.00001  0.00001
430=
440=
450=
460=
470=
480=EOF
490=EOF

```



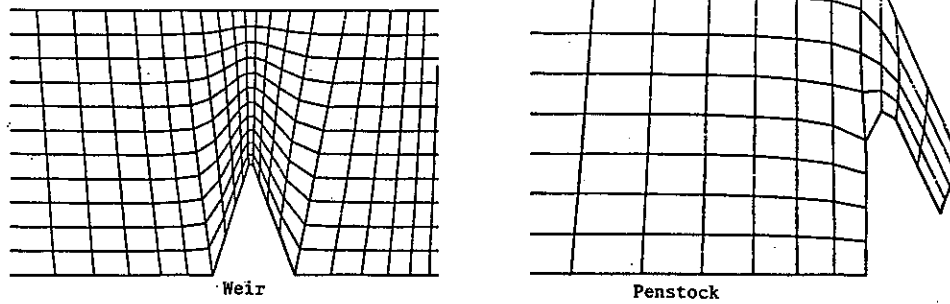
WESCOR System #1

(Reduced Horizontal Scale)



WESCOR System #2

(Reduced Horizontal Scale)



In accordance with letter from DAEN-RDC, DAEN-ASI dated 22 July 1977, Subject: Facsimile Catalog Cards for Laboratory Technical Publications, a facsimile catalog card in Library of Congress MARC format is reproduced below.

Thompson, Joe F.

A boundary-fitted coordinate code for general two-dimensional regions with obstacles and boundary intrusions / by Joe F. Thompson (Department of Aerospace Engineering, Mississippi State University). -- Vicksburg, Miss. : U.S. Army Engineer Waterways Experiment Station ; Springfield, Va. ; available from NTIS, 1983.

80 p. : ill. ; 27 cm. -- (Technical report ; E-83-8)

Cover title.

"March 1983."

Final report.

"Prepared for Office, Chief of Engineers, U.S. Army under EWQOS Task IIIA.4."

"Monitored by Hydraulics Laboratory, U.S. Army Engineer Waterways Experiment Station."

At head of title: Environmental & Water Quality Operational Studies.

Bibliography: p. 48.

Thompson, Joe F.

A boundary-fitted coordinate code for general : ... 1983.  
(Card 2)

1. Boundary value problems. 2. Computer programs.  
3. Coordinates. 4. Difference equations. 5. Differential equations. 6. Finite difference equations. 7. Numerical analysis. I. Mississippi State University. II. United States. Army. Corps of Engineers. Office of the Chief of Engineers. III. Environmental & Water Quality Operational Studies. IV. U.S. Army Engineer Waterways Experiment Station. Hydraulics Laboratory. V. Title VI. Series: Technical report (U.S. Army Engineer Waterways Experiment Station) ; E-83-8.  
TA7.W34 no.E-83-8